A SENSITIVITY ANALYSIS OF A BIOLOGICAL MODULE DISCOVERY PIPELINE

By

James Long

RECOMMENDED: ______________________________
Dr. Chris Hartman

____
Dr. Thomas Marr

____
Dr. John Rhodes

____
Dr. Mitchell Roth
Advisory Committee Chair

____
Dr. Jon Genetti
Chair, Department of Computer Science

APPROVED: ______________________________
Dr. Douglas Goering
Dean, College of Engineering and Mines

____
Dr. John Eichelberger
Dean of the Graduate School

24 April 2015
____
Date

A SENSITIVITY ANALYSIS OF A BIOLOGICAL MODULE DISCOVERY PIPELINE

A

DISSERTATION

Presented to the Faculty

of the University of Alaska Fairbanks

In Partial Fulfillment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

By

James Long, M.S., B.S.

Fairbanks, Alaska

May 2015

Abstract

Gene expression is the term applied to the combination of **transcription**, the process of copying information stored in DNA (deoxyribonucleic acid) into a **transcript**, and **translation**, the process of reading a transcript in order to manufacture a cellular product. Cellular products are typically proteins, which can combine either structurally or in concert to accomplish one or more tasks. Cooperating protein combinations are called **modules**, and it is thought that groups of transcripts with high correlation between their respective concentrations may indicate such modules.

An open-source version of the CODENSE algorithm was developed with improved correlation methods to computationally test this hypothesis on an artificial transcription network containing a known module motif. The artificial network was used as input to a biochemical simulator in order to obtain synthetic transcription data, which was then fed to the pipeline whose purpose it is to discover modules in such data. Any discovered modules are compared to the known modules in the original network during a sensitivity analysis, where the process is repeated thousands of times with slightly varied parameters for each run. This process quantifies the sensitivity of pipeline output to each parameter of the pipeline, the most sensitive of which suggest what parts of the pipeline may be candidates for further refinement. The sensitivity analysis was then extended to include variation of biological network parameters, and noisy data. Lessons learned were then extended to the case of two known modules.

Dedication

I would like to dedicate this project to family and friends, without whom I would not have finished.

First, I would like to thank Dr. Mitchell Roth, mentor and friend for 20 years. I will always be inspired by his dedication to students, and attention to detail.

I would also like to thank my wife Merced for her love and support. She is truly my better half.

Finally, I would like to thank my father, James Long Sr., who passed away just 2 months before this project was finished. His faith in God will always be remembered.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Appendices

0: Prolegomena: Gene Expression for Computer Scientists

0.1 Introduction

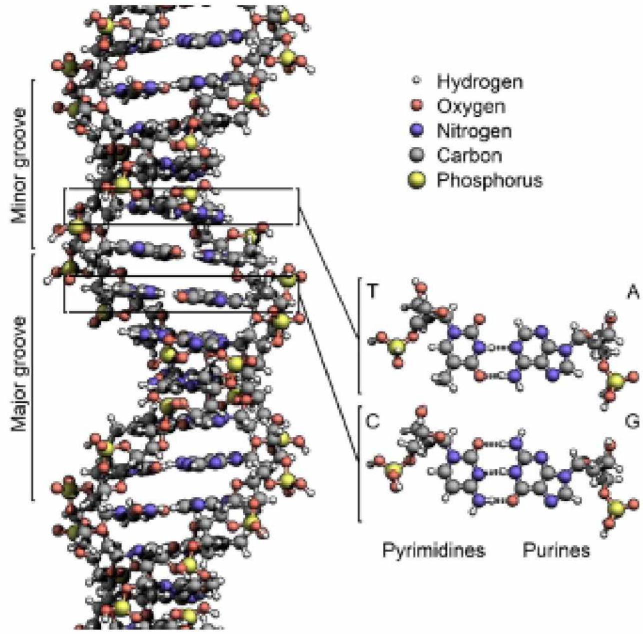


Fig. 0.1: DNA

The term 'gene expression' refers to the process of reading information (transcription) stored in DNA (deoxyribonucleic acid, Fig. 0.1 [Zephyris, 2011]) in order to manufacture a cellular product (translation). Cellular products are typically proteins, which can combine either structurally or in concert to accomplish one or more tasks. Such protein combinations are called 'modules', and it is thought that high expression correlation amongst the precursors to protein products may indicate such modules [Zhou et al., 2005]. The object of this dissertation is to investigate this claim, beginning with an introductory treatment of gene expression intended to provide background and context for what is to follow. Terms and concepts from Computer Science are employed alongside those from the biological nomenclature.

0.2 Transcription

DNA is a two-stranded string over an alphabet of four letters, each of which is the initial letter of one of the four nucleotide (or base) molecules that are the primary structural units of DNA, viz Adenine, Cytosine, Guanine, and Thymine.

Within the interior of the double helix, letters of each strand are linearly arranged along their respective backbone (the two outer edges of the double helix), with string lengths ranging from roughly 10^6 to 10^9 [Alon, 2006]. Each letter of one strand weakly binds to a letter on the opposite strand, i.e. As on one strand pair with Ts on the other strand, and Cs pair with Gs. Every such pair is known as a base-pair, abbreviated bp.

Contiguous strings are known as chromosomes, the length and number of which are species dependent. Within eukaryotic cells, i.e. those that contain a membrane bound nucleus, chromosomes can exhibit extremely complex tertiary and quaternary structure. DNA is spooled around numerous molecules known as histones, which are incorporated into successive rounds of scaffolding proteins responsible for the appearance of chromosomes in the microscope. Eukaryotes are typified by complex organisms such as animals, plants, and fungi, in opposition to prokaryotes, cells without a nucleus, typified by the bacteria and archaea. Chromosomal structure is not as complex in prokaryotes. Regardless of the organism, it is sufficient for this study to consider only the base-pair strings themselves, illustrated by the abstraction to be used hereafter in Fig. 0.2.

TATATACATGCCGTATGTATGCGTATATGCGTACGTCGATGCAG
ATATATGTACGGCATACATACGCATATACGCATGCAGCTACGTC

Fig. 0.2: Base-pair strings

A small percentage of DNA consists of substrings known as genes, each of which contain one or more coding regions, known as exons, which may be separated by non-coding regions, known as introns. Most genes may be conceptualized as the source code for a base-class protein, and related subclasses. Not all genes code for proteins; some code for other parts of the molecular protein assembly line composed of various forms of RNA (ribonucleic acid.) Genes vary in length from roughly 10^3 to 10^6 bp [Alon, 2006].

During transcription, the coding region will be parsed by an enzyme known as RNAP (ribonucleic acid polymerase), the first step in preprocessing the source code [Ptashne, 1992]. The much larger RNAP is assisted in its initial positioning with the help of transcription factors, smaller protein complexes that bind to specific sequences in the DNA (binding sites) that are contained within promoters, regions of DNA that precede a gene, Fig. 0.3.

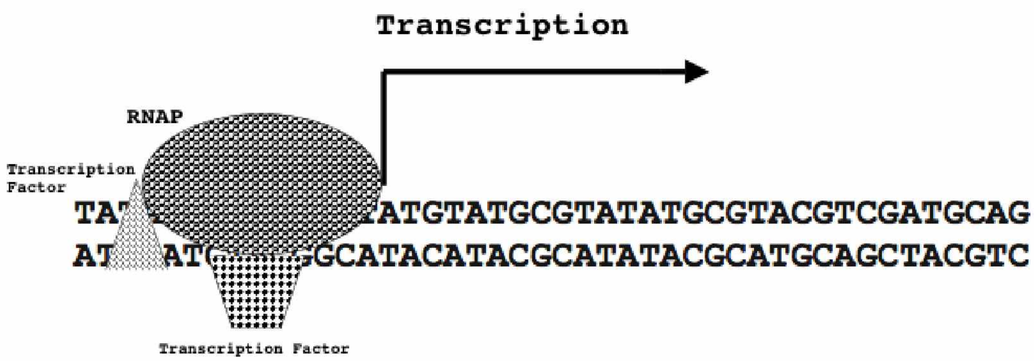


Fig. 0.3: Transcription factors and RNAP

Transcription factors that increase transcription rates are known as activators, while others that can decrease rates, or even physically block transcription are known as repressors, Fig. 0.4.

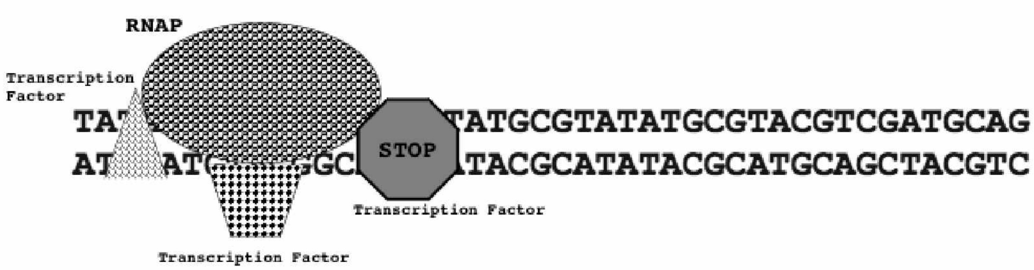


Fig. 0.4: A repressor transcription factor

As the RNAP parses the protein coding region, it manufactures a single-stranded copy of the sense strand of the coding region in the form of mRNA (messenger RNA), except that each nucleotide Thymine in the DNA is replaced with a Uracil nucleotide in the mRNA, Fig. 0.5. The strand not copied is the antisense strand.

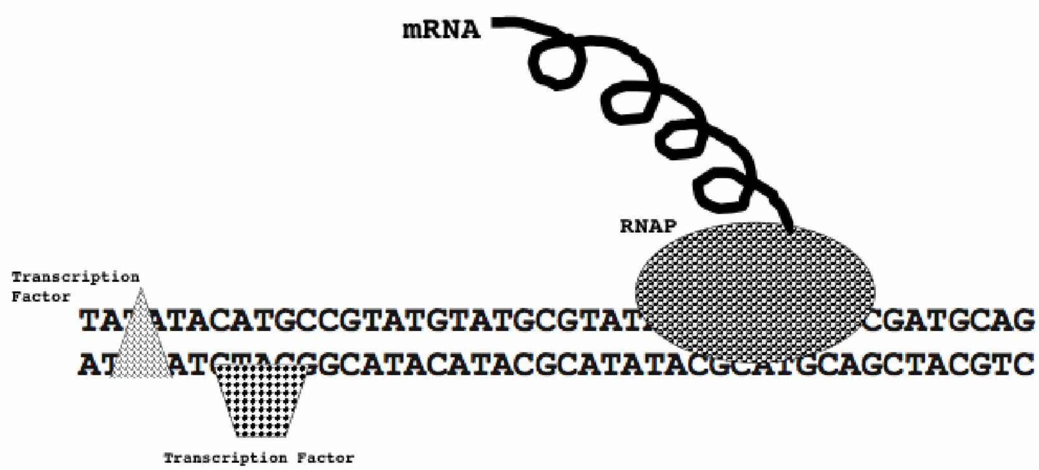


Fig. 0.5: mRNA strand production

At the conclusion of transcription, the manufactured mRNA strand detaches from the RNAP to undergo post-transcriptional modification (source code preprocessing), and translation (compilation) to a protein. Modification of the mRNA may include splicing, where introns are excised, editing, which changes the mRNA nucleotide composition, and polyadenylation, the addition of a multiple As to the end of the the mRNA known as a poly(A) tail. During high rates of transcription, additional RNAP may be recruited to the DNA binding site by the transcription factors before the previous RNAP detaches from the DNA, Fig. 0.6.

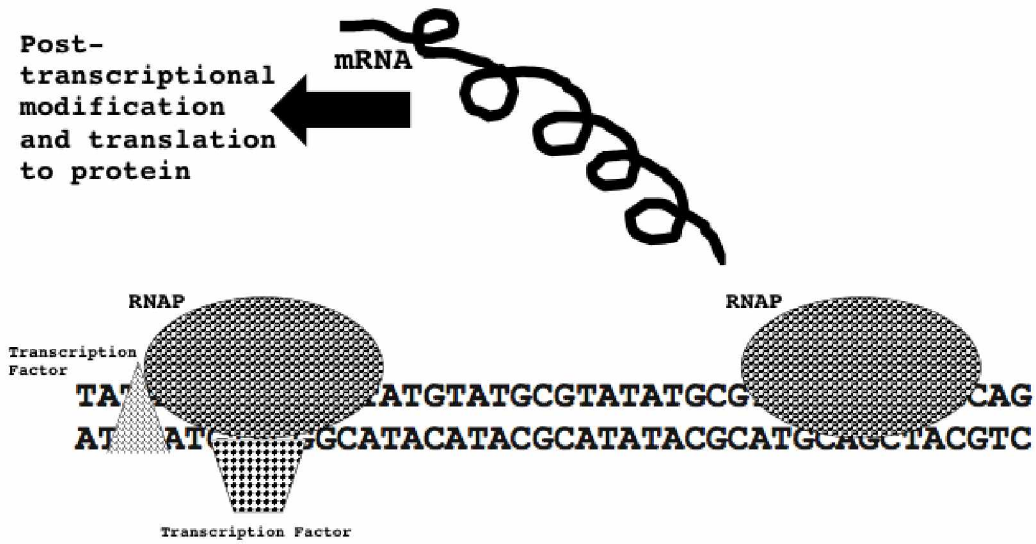


Fig. 0.6: Completion of transcription

Throughout what is to follow, it is important to remember that the transcription factors themselves are products of gene expression, and will appear in equations for gene expression rates.

0.3 Translation

Once the preprocessing of mRNA during post-transcriptional modification is accomplished, translation of the mRNA into a protein may commence. If this protein will be part of some larger complex, the translation is analogous to the compilation of source code into a library. If the protein will act independently, the translation is analogous to the compilation of source code into a binary.

In a eukaryote, the mRNA is transported outside the nucleus, where a ribosome can translate, or compile, it into a protein. The rules of translation are guided by the genetic code, which defines a mapping from each set of three consecutive nucleotides in the mRNA, known as a codon, to an amino acid in the protein.

Since there are four different nucleotides that may comprise the elements of a codon, there are thus $4^3=64$ possible codons. The genetic code maps 61 of these different codons onto 20 possible amino acids, as shown in Table 0.1. The other 3 codons are 'Stop' codons, indicating that that translation should cease.

The protein is manufactured as a string of amino acids in the same order in which their respective codons appear in the mRNA. As this string exits the translation machinery, it begins to fold into its functional shape due to a combination of forces between the constituent amino acids, and forces from the environment. Translation terminates when a 'Stop' codon is encountered.

Table 0.1: The genetic code

		2nd base							
		A		**C**		**G**		**U**	
	A	**AAA**	Lys/K	**ACA**	Thr/T	**AGA**	Arg/R	**AUA**	Ile/I
		AAC	Asn/N	**ACC**	Thr/T	**AGC**	Ser/S	**AUC**	Ile/I
		AAG	Lys/K	**ACG**	Thr/T	**AGG**	Arg/R	**AUG**	Met/M
		AAU	Asn/N	**ACU**	Thr/T	**AGU**	Ser/S	**AUU**	Ile/I
	C	**CAA**	Gln/Q	**CCA**	Pro/P	**CGA**	Arg/R	**CUA**	Leu/L
		CAC	His/H	**CCC**	Pro/P	**CGC**	Arg/R	**CUC**	Leu/L
1st base		**CAG**	Gln/Q	**CCG**	Pro/P	**CGG**	Arg/R	**CUG**	Leu/L
		CAU	His/H	**CCU**	Pro/P	**CGU**	Arg/R	**CUU**	Leu/L
	G	**GAA**	Glu/E	**GCA**	Ala/A	**GGA**	Gly/G	**GUA**	Val/V
		GAC	Asp/D	**GCC**	Ala/A	**GGC**	Gly/G	**GUC**	Val/V
		GAG	Glu/E	**GCG**	Ala/A	**GGG**	Gly/G	**GUG**	Val/V
		GAU	Asp/D	**GCU**	Ala/A	**GGU**	Gly/G	**GUU**	Val/V
	U	**UAA**	Stop	**UCA**	Ser/S	**UGA**	Stop	**UUA**	Leu/L
		UAC	Tyr/Y	**UCC**	Ser/S	**UGC**	Cys/C	**UUC**	Phe/F
		UAG	Stop	**UCG**	Ser/S	**UGG**	Trp/W	**UUG**	Leu/L
		UAU	Tyr/Y	**UCU**	Ser/S	**UGU**	Cys/C	**UUU**	Phe/F

Legend						
	Ala/A	Alanine	Gly/G	Glycine	Pro/P	Proline
	Arg/R	Arginine	His/H	Histidine	Ser/S	Serine
	Asn/N	Asparagine	Ile/I	Isoleucine	Thr/T	Threonine
	Asp/D	Aspartic acid	Leu/L	Leucine	Trp/W	Tryptophan
	Cys/C	Cysteine	Lys	Lysine	Tyr/Y	Tyrosine
	Gln/Q	Glutamine	Met/M	Methionine	Val/V	Valine
	Glu/E	Glutamic acid	Phe/F	Phenylalanine		

0.4 Expression Rate

Transcription factors exist in one of two states, either inactive (unable to bind to DNA), or active (capable of binding to DNA). Transition from an inactive to active state (an allosteric conformational change) occurs rapidly in response to signals received from the cellular environment, thus affecting the rate of production of those proteins that can then act on the environment in response to the signal. Since transcription factors are themselves proteins, their rate of production is regulated by other transcription factors, whose production rate is also regulated by other transcription factors, and so on. This transcription network of interactions is the system responsible for responding to changes in the cellular environment.

Changes to transcription factor activation states occurs over sub-second intervals, while DNA binding of transcription factors occurs over intervals of seconds, and production of sufficient amounts of protein product occurs over minutes to hours [Alon, 2006]. In light of this difference in time scales, a first order approximation is made that assumes steady-state transcription factor activation concentrations for protein production rates described by equations known as Hill Functions, described next.

Assume that gene expression rate $f(x)$ for a protein P is controlled by a single activator transcription factor with concentration G_0. Letting $x=G_0/K$, where K (the activation coefficient) represents the activator concentration level that turns a gene on, one can imagine a gene expression rate for P modeled as a step function, with the gene turned off for activator concentrations of x below 1, and on, with expression rate B, for concentrations above 1.

In reality, there is a gradual change in expression rate for P, modeled by the Hill Function (Eqn. 0.1), where the Hill coefficient n controls the slope of the step function, see Fig. 0.7.

$$f(x)=\frac{Bx^n}{1+x^n}, x\geq 0 \qquad \textit{Eqn. 0.1}$$

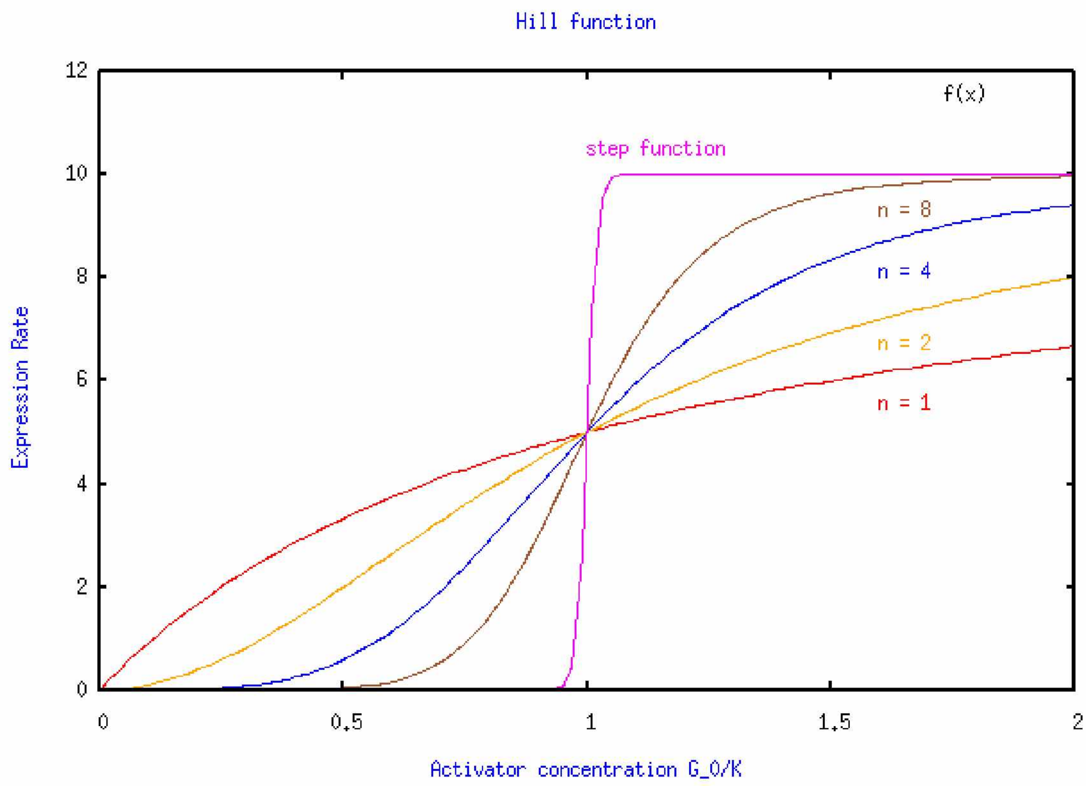


Fig. 0.7: Hill Function expression rate for one activator transcription factor

At $x=1$, $f(x)=\frac{B}{2}$ for all values of n.

For a single repressor transcription factor, the Hill Function graph is inverted, Fig. 0.8, described by Eqn. 0.2.

$$f(x)=\frac{B}{1+x^n} \qquad \textit{Eqn. 0.2}$$

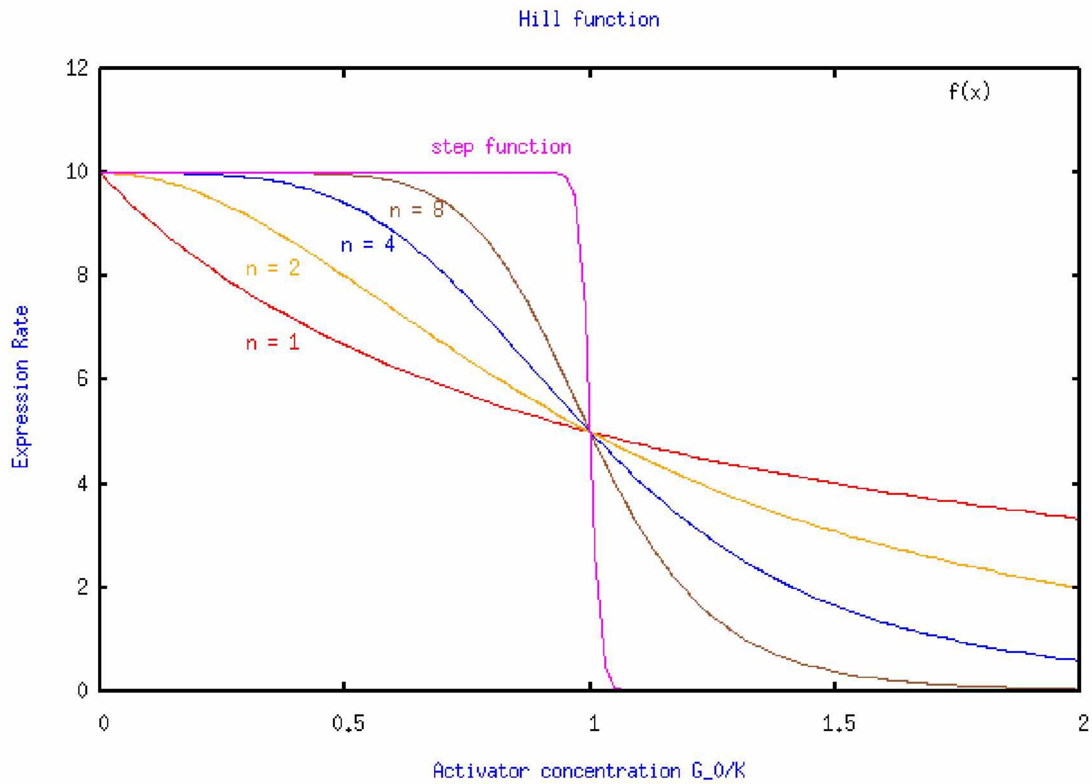


Fig. 0.8: Hill Function expression rate for one repressor transcription factor

Assuming that transcription factors bind independently without interaction among themselves, the gene expression rate for a protein controlled by multiple such factors may be modeled by the Generalized Hill Function

$$f(\bar{x}) = \frac{\sum_i B_i x_i^{n_i}}{1 + \sum_i x_i^{m_i}}, \quad n_i = m_i \, for\ activators, \quad n_i = 0, m_i > 0 \, for\ repressors \qquad Eqn.\ 0.3$$

This equation is central to the modeling efforts that follow, where a computer generated transcription network containing known modules is fed to the COPASI [Hoops et al., 2006] biochemical simulator in order to obtain synthetic gene expression data. This data is then fed to a pipeline of algorithms whose purpose is to discover modules in such data, and any discovered modules are compared to the known modules in the original network via sensitivity analysis, where the process is repeated thousands of times with slightly varied parameters for each run. This analysis quantifies the sensitivity of pipeline output to each parameter of the pipeline, the most sensitive of which suggest what parts of the pipeline, if any, may be candidates for further refinement. The analysis is then extended to include variation of biological network parameters.

1: Gene Expression Data

High expression correlation amongst a subset of the protein precursor products (mRNA) suggests the existence of a biological module, viz the elements of that subset code for proteins that participate in a common function [Zhou et al., 2005]. Such data is typically obtained from microarrays or Next Generation Sequencing (NGS) machines, where the topology of the networks that emit the data is usually unknown. In this chapter, a method of generating gene expression data from a known artificial transcription network is described, which will be used as input to the module discovery pipeline under consideration.

1.1 Synthetic Data

Known transcription networks of sufficient size and variability facilitate objective testing and comparison of network deconvolution algorithms, and can be used for sensitivity analysis. For several years now, data of this type has been available [Mendes et al., 2003] as a set of synthetic networks consisting of 100 genes and 200 interactions organized in two ways, as Erdös-Rényi random networks, and as scale-free topology networks. Meanwhile, however, more work has been done on what real biological networks look like, aptly summarized as a series of network motifs [Alon, 2006], where examples are also given of how the various network motifs are topologically generalized into larger structures, along with their connection hierarchy.

The method to be described here for generating synthetic microarray data relies upon formation of an artificial transcription network based on these known motifs. The Random Network Generator (RANGE [Long & Roth, 2008]) can generate large random transcription networks, with up to 16,000 genes, in the Network Motif (NEMO [Long & Roth, 2008]) language, producing a "fat-tail" distribution, whose higher degree nodes follow a power-law. NEMO's compiler (nemo2sbml) uses lex and yacc [Johnson, 1979] to output a Systems Biology Markup Language (SBML [Hucka et al., 2004]) model for either user-specified and/or randomized gene input functions. The randomized gene input functions are Generalized Hill Functions [Alon, 2006, Likhoshvai & Ratushny, 2007], whose parameters are picked randomly from a uniform distribution. Importing an SBML model of a known network into a biochemical simulator allows the generation of synthetic microarray data for algorithm development purposes. The software distribution includes a script in the R language [Ihaka & Gentleman, 1996] that can add noise to the synthetic data.

The current default behavior is that all transcription factors are active. In order to simulate the allosteric effects of signal transduction, groups of transcription factors may be turned on and off by editing the SBML model's XML appropriately, to be described later.

1.2 The Network Motif (NEMO) Language

The NEMO language will be introduced by way of example:

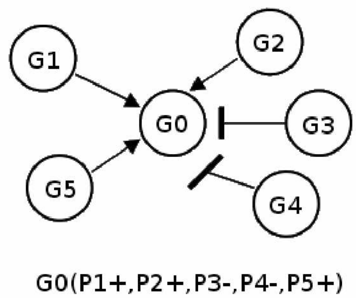
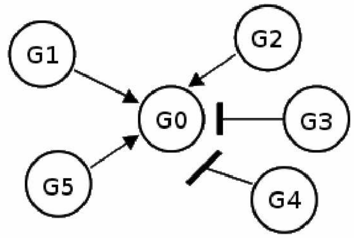


Fig. 1.1: G0 *transcription factors*

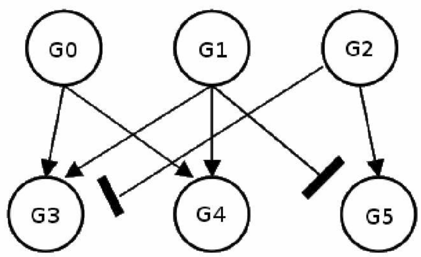
In Fig. 1.1, gene G0 is regulated by products from five other genes: up-regulated by G1, G2, and G5, and down-regulated by genes G3 and G4. This relationship is expressed in the NEMO language as G0(P1+,P2+,P3-,P4-,P5+), where the letter P may be thought of as standing for the "product" of a gene, often a protein. A complete description of the network using this non-motif syntax would include a list of such statements, one for each gene as arguments to a GLIST (gene list) construct, enclosed by square brackets which mark the beginning and ending of a network description, as illustrated in Fig. 1.2.




```
[GLIST(G0(P1+,P2+,P3-,P4-,P5+),
       G1(...),
       G2(...),
       etc.  )]
```

Fig. 1.2: NEMO expression for G0 *set in a network description*

While any network could be delineated in the manner just described, NEMO has language constructs for network motifs as categorized in [Alon, 2006], illustrated by the following examples:




```
DOR(G3(P0+,P1+,P2-),
    G4(P0+,P1+)
    G5(P1-, P2+))
```

Fig. 1.3: A Dense Overlapping Regulon (DOR) motif

Fig. 1.3 depicts a Dense Overlapping Regulon (DOR) motif, which is a connected bipartite graph with transcription factors on the top, and regulated genes on the bottom. For this network motif, the NEMO statements for each gene are as before, but are now arguments to a DOR construct instead of a GLIST construct.

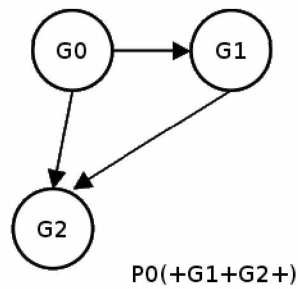


Fig. 1.4: A Feed-Forward Loop (FFL) motif

Fig. 1.4 illustrates a Feed-Forward loop (FFL) motif, where a product from G0 up-regulates G1, and a product from G1 up-regulates G2, which is also up-regulated by a product from G0.

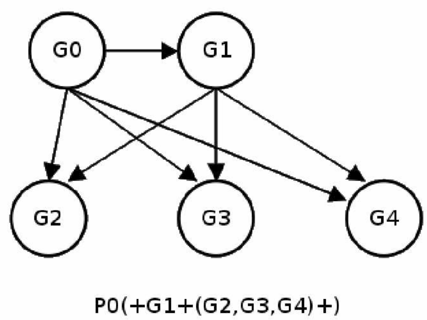


Fig. 1.5: A multi-output FFL motif

Fig. 1.5 illustrates a multi-output FFL motif, where G0 up-regulates G1, G1 up-regulates G2, G3, and G4, all three of which are also up-regulated by G0.

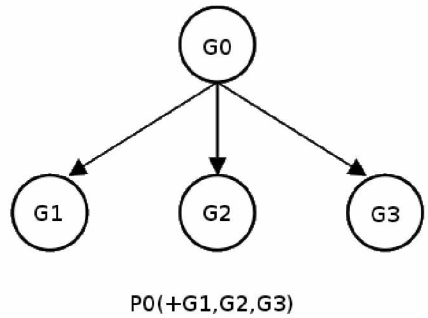


Fig. 1.6: A Single-Input module (SIM) motif

Figure 1.6 illustrates a Single-Input Module (SIM) motif, where G0 up-regulates G1, G2, and G3. This simple motif is employed several times in the synthetic network used in this study, where it is assumed that the genes are expressed in roughly equal quantities (see section 5.2).

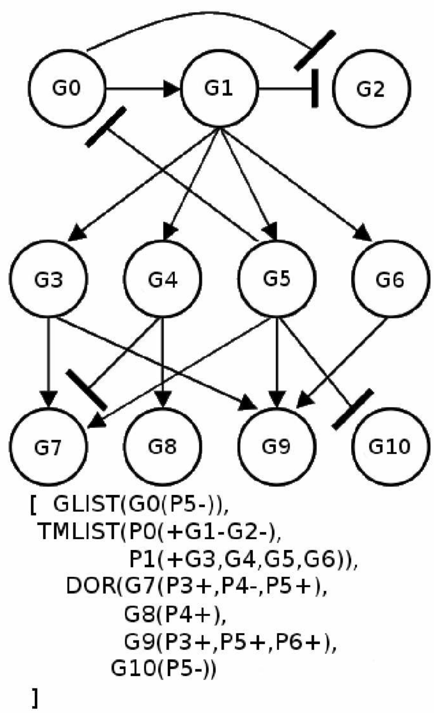


Fig. 1.7: A complete NEMO network

Figure 1.7 illustrates a complete network, indicated by enclosing square brackets. Motifs other than the Dense Overlapping Regulon (DOR) are grouped into a Transcription Motif List (TMLIST), composed of a Feed-Forward Loop (FFL) and a Single-Input Module (SIM).

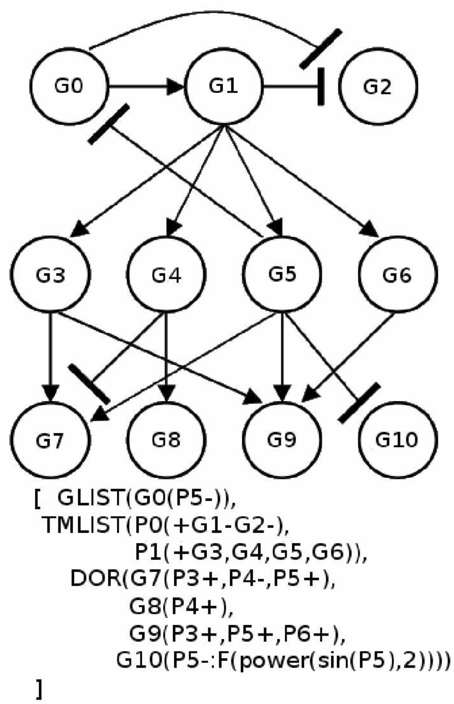


Fig. 1.8: A complete NEMO network with a specified input function

Figure 1.8 illustrates the same complete network as figure 1.7, except that the default Hill Function for G10 is replaced by a specified function.

1.3 Data Mechanics

1.3.1 RANGE/NEMO Installation

Download the code from http://range.sourceforge.net, then at a shell command prompt

```
$ tar zxf range_nemo-1.6.tar.gz
$ cd range_nemo-1.6
```

and install according to the directions in README.txt. One of the products of a successful installation is the NEMO to SBML compiler, nemo2sbml.

1.3.2 Data Generation

Given a text file named 'network.txt' describing a network in the NEMO language, constructed either by RANGE or by hand, synthetic microarray data is generated by first compiling the NEMO language description into a SBML representation, and optionally renaming the resulting SBML file:

```
$ ./nemo2sbml network.txt sbml100
$ mv sbml100_0.xml sbml100.xml
```

For genes whose input function is not explicitly defined, the compiler inserts Generalized Hill Functions with random parameters before translation of the network into SBML format. Randomized parameters mean that most genes, usually all, are turned 'on', meaning that all transcription factors are active, which is rarely the case. In reality, transcription factors are generally inactive, meaning they cannot bind to DNA, and only become active by changing their shape in response to signals from outside the cell. This allosteric change in the conformation of a transcription factor allows it to bind to DNA, and thus affect the necessary genes needed to respond to the signal.

In order to produce data sets that mimic allosteric signal transduction, the SBML file generated above is imported into the biochemical simulator, and saved in its proprietary format. A template is produced from this proprietary format by editing it to turn off all the gene expression functions, accomplished by setting every B in the Hill Functions to zero. When data needs to be generated, only those transcription factors needed are turned on in the template before ingestion by the biochemical simulator. The mechanics of this are as follows:

Start the COPASI biochemical simulator [Hoops et al., 2006], version 4.8 (Build 35), and

1) under the 'File' menu, click 'Import SBML', and import 'sbml100.xml'
2) expand 'Tasks' in the left pane, select 'Time Course'
3) set 'Duration' to 8 s, 'Intervals' to 800, and 'Interval Size' to 0.01 (See section 5.2)
4) check the 'executable' box
5) click the 'Output Assistant', and create report 'Complete Time, Concentrations, Volumes, and Global Quantity Values'
6) click the 'Report' button, enter 'CopasiRun.txt' as the target for the above report, uncheck 'Append', and confirm
7) under the 'File' menu, click 'Save' and save as 'sbml100.cps' (.cps is the proprietary COPASI XML format)

The template with all Hill Functions turned off, 'sbml-template.cps', is created from 'sbml100.cps' by the following command:

```
$ cat sbml100.cps |
sed "s/\(name=\"B_[0-9]*\"\) value=\".*\"/\1 value=\"0.0\"/"   >   sbml-
template.cps
```

the results of which should be checked with

```
$ cat sbml-template.cps | grep "name=\"B_[0-9]*\" value=\".*\"" | more
```

to verify that every B has indeed been set to zero.

After the template is modified to turn on a subset of the transcription factors, synthetic microarray data is generated by supplying the modified template as an argument to the command line version of the COPASI biochemical simulator. Template modification is handled by shell and python scripts described in detail in chapter 6.

1.3.3 Preliminary Testing

In the Prolegomena, the Hill Function was qualitatively plotted, Figure 0.7, as gene 'Expression Rate' vs. 'Activator concentration G_0/K', where G_0 is the concentration of gene product, and K is the concentration value of its activator that turns expression on. The COPASI simulator solves a differential equation for expression concentration, where the derivative of the concentration with respect to time is equal to the Hill Function minus a degradation coefficient times the concentration. In order to verify the simulator for proper behavior, a plot of concentrations will be made for two genes, where one is turned on by the other, whose concentration increases linearly over time. To this end, we start with the simple NEMO network

```
[ GLIST(G0(P1+),
        G1(P0+))
]
```

and save it in a file called 'small'. The resulting SBML will be modified, and a plot made via the following:

1) compile 'small' with nemo2sbml, and rename the resulting file 'small.xml':

```
$ ./nemo2sbml small
$ mv regulatoryNetwork_2genes_0.xml small.xml
```

2) start the COPASI biochemical simulator [Hoops et al., 2006], version 4.8 (Build 35)

```
$ ~/copasi/bin/CopasiUI &
```

and

a) under 'File' menu, click 'Import SBML', and import 'small.xml'
b) expand 'Tasks' in the left pane, select 'Time Course'
c) set 'Duration' to 10 s, 'Intervals' to 1000, and 'Interval Size' to 0.01
d) check the 'executable' box
e) click the 'Output Assistant', and create 'Reports'-->'Complete Time, Concentrations, Volumes, and Global Quantity Values'
f) click the 'Report' button, enter 'small.txt' as the target for the above report, uncheck 'Append', and confirm
g) under 'File' menu, save in the proprietary COPASI XML format as 'small.cps', and close

3) turn off all genes in 'small.cps', and save the result as a template:

```
$ cat small.cps | sed "s/\(name=\"B_[0-9]*\"\) value=\".*\"/\1
value=\"0.0\"/" > small-template.cps
```

4) to ensure that P0 increases with time, change the P0 synthesis function in the template to a positive constant derivative with no degradation term by replacing

```
B_0*(P1/K_0)^n_0/(1+(P1/K_0)^n_0)/tau/Cell
```

with

```
B_0/tau/Cell
```

and also change

```
<Constant key="Parameter_82" name="B_0" value="0.0"/>
```

to

```
<Constant key="Parameter_82" name="B_0" value="1.0"/>
```

```
P0/tau/Cell
```

to

```
0.0*P0/tau/Cell
```

and

```
<Constant key="Parameter_78" name="dc_0" value="0.0152468"/>
```

to

```
<Constant key="Parameter_78" name="dc_0" value="0.0"/>
```

Turn the expression rate on sharply (high *n*) by changing

```
<Constant key="Parameter_75" name="B_1" value="0.0"/>
```

to

```
<Constant key="Parameter_75" name="B_1" value="10.0"/>
```

```
<Constant key="Parameter_73" name="n_1" value="1"/>
```

to

```
<Constant key="Parameter_73" name="n_1" value="10"/>
```

and

```
<InitialState type="initialState">
  0 6.02214179e+23 6.02214179e+23 0 1
```

to

```
<InitialState type="initialState">
  0 0 0 0 1
```

5) save as 'small-HillFunc.cps', and open in COPASI simulator

```
$ ~/copasi/bin/CopasiUI small-HillFunc.cps &
```

6) expand 'Tasks' in the left pane, select 'Time Course'

7) click the 'Output Assistant', and create 'Plots'-->'Complete Concentrations, Volumes, and Global Quantity Values'

8) hit 'Run', and deselect all plots but P0 and P1, see Fig. 1.9

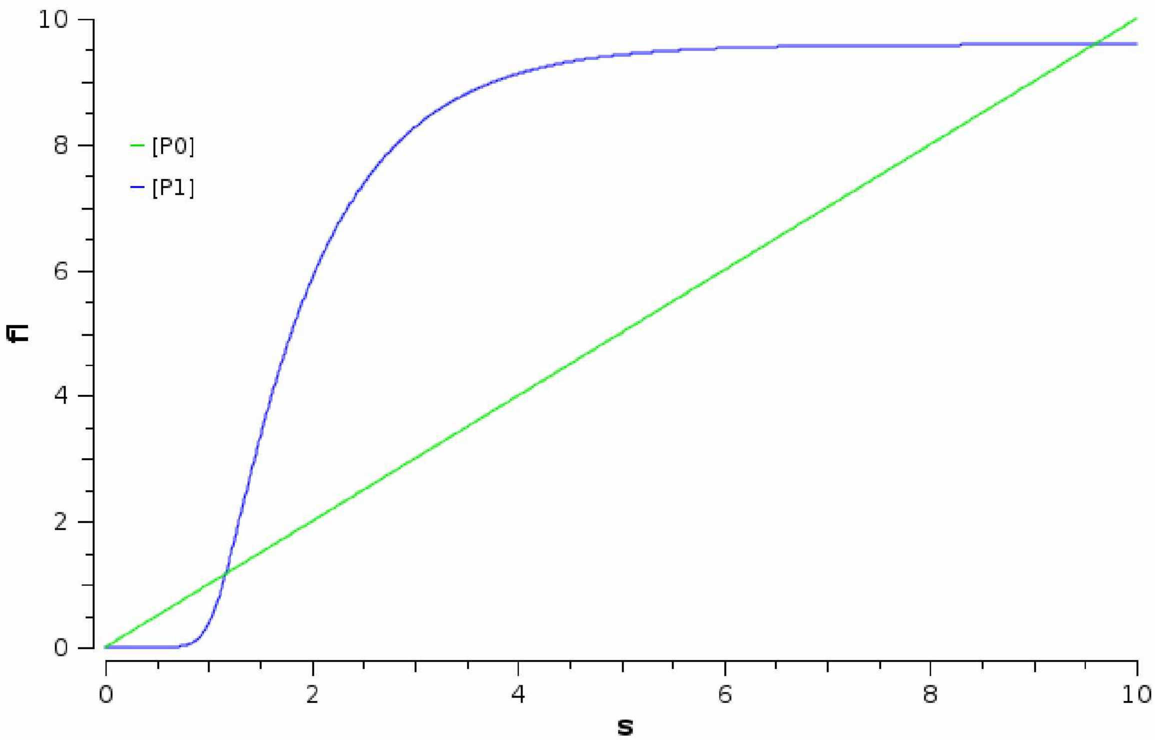


Fig. 1.9: COPASI output for a simple two gene network

This output, plotted as femtoliters vs seconds, looks as expected, with P1 suppressed until P0 reaches sufficient concentration to turn P1 on. The rate of ascent for P1 is controlled by the Hill Function and degradation, with steady-state occurring roughly after the 4 second mark. Typical time-to-steady-state values are important to know for sensitivity analysis runs, and are examined in section 5.2.

2: Dense Subgraph Algorithm

A critical component of the software architecture discussed in the next chapter is a method to determine the density of subgraphs in a graph that exceed or equal a specified minimum, even if they overlap. An algorithm was developed as part of this research (**ODES**: Overlapping DEnse Subgraphs [Long & Hartman, 2010]) that finds all overlapping dense subgraphs with density of at least 0.5, based on the following theorem:

2.1 Theorem

A connected graph *G*, with *e* edges, $n \geq 3$ vertices, and density $den(G) \geq 0.5$, has at least one non-cut vertex *w* whose degree $d(w)$ is less than or equal to the average degree d_{ave} of vertices in *G*, i.e. $d(w) \leq d_{ave} = \frac{2e}{n}$. Removal of *w* from *G* does not decrease the density of *G*.

Proof

Given a connected graph *G*, with *e* edges, $n \geq 3$ vertices, and $den(G) \geq 0.5$, pick some vertex *v* where $d(v) \leq d_{ave}$. If *v* is not a cut vertex, then *w* = *v*. If *v* is a cut vertex, then let *S* be the smallest component of $G - v$, and let $A = S + v$ & $B = G \setminus A$.

Setting $x = |V(S)|$ and $y = |V(B)|$, we then have $x \leq y$, $n = x + y + 1$, and $|V(A)| = x + 1$. See Fig. 2.1.

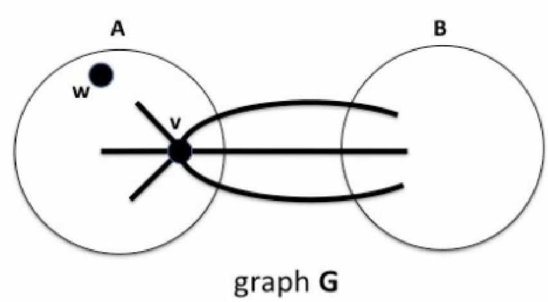


*Fig. 2.1: A graph G with cut vertex **v***

It is well known that there exist at least two non-cut vertices in any connected graph with more than one vertex, and thus there must be at least one non-cut vertex w for A that is different from v. Since w is not a cut vertex in A, it is not a cut vertex in G.

To show that $d(w) \leqslant d_{ave} = \frac{2e}{n}$, we note that

$$den(G) \geq \frac{1}{2} \quad \Rightarrow \quad \frac{2e}{n(n-1)} \geq \frac{1}{2} \text{ , or}$$

$$d_{ave} \geq \frac{n-1}{2} = \frac{x+y}{2} \quad .$$

Now $d(w) \leq |v(A)| - 1 = x = \frac{x+x}{2} \leq \frac{x+y}{2} \leq d_{ave}$.

Removal of w from G does not decrease the density of G:

$$d(w) \leq \frac{2e}{n} \quad \Rightarrow \quad nd(w) \leq 2\mathrm{e} \quad \Rightarrow$$

$$nd(w) - ne \leq 2\mathrm{e} - ne \quad \Rightarrow$$

$$n(e - d(w)) \geq e(n-2) \quad \Rightarrow$$

$$\frac{e - d(w)}{n-2} \geq \frac{e}{n} \quad \Rightarrow$$

$$\frac{2(e - d(w))}{(n-1)(n-2)} \geq \frac{2e}{n(n-1)} \quad \Rightarrow \quad den(G - w) \geq den(G).$$

Q.E.D.

The theorem says that vertices can iteratively be removed from a sufficiently dense graph, without decreasing its density or cutting it into disconnected pieces, until all that remains is a single pair of connected vertices. The algorithm goes the other way: Starting with the set S of all connected pairs of vertices (single-edge subgraphs), an iteration consists of looking for adjacent vertices that can be added to each member of S consistent with the theorem.

A member S_i of S to which a vertex can be added is placed with the added vertex into a sorted list L for the next iteration, one for each new vertex that can be added to S_i . The brute-force search space of a non-clique graph G is thus confined to the actual dense subgraphs of G, and since each iteration is independent, it can be handled by its own thread. During any iteration, a dense subgraph is saved for output if it is of sufficient size, and no more vertices can be added.

More specifically, any maximal subgraph with density above a specified minimum $D \geq 0.5$ can be found in the following manner:

A list S is initialized with all single-edge subgraphs S_i of G. Then, for each S_i in S, a search is made for every vertex v adjacent to S_i whose degree in $S_i + v$ is less than or equal to the average degree of $S_i + v$, and whose addition to S_i maintains the density of $S_i + v$ above D, i.e. a search is made for every v that satisfies $den(S_i) \geq den(S_i + v) \geq D \geq 0.5$.

For each v that qualifies, $S_i + v$ is inserted into a sorted list L, provided that $S_i + v$ is not already in L, determined by a binary search. When no other entries can be added to L, the contents of L become the new S, and another iteration with an initially empty L commences. If no entries have been made into L, the process is terminated.

The list of dense subgraphs R to return is built by appending any S_i to R that is not extended within an iteration, assuming its order (number of vertices) exceeds some specified minimum.

Strictly implementing the algorithm in conformance with the theorem requires that the current density of each S_i be tracked in order to ensure that $den(S_i) \geq den(S_i + v)$. This requirement may be dropped, however, by noting that any $S_i + v$ whose density is less than $den(S_i)$, but greater than D, is really a pre-computation of some other subgraph S_k with a different adjacent vertex w added consistent with the theorem, i.e., $S_i + v$ can be reordered as $S_k + w$, $v \neq w$, by our theorem so that $den(S_k) \geq den(S_k + w) \geq D \geq 0.5$, and may thus be added to L "early", assuming it is not already in L. It is sufficient then to insure only that $den(S_i + v) \geq D \geq 0.5$, eliminating the need to track current densities.

Each round of extending a set of candidate subgraphs S_i by one vertex is independent of previous rounds, and so can be handled by a separate thread of execution. Candidate subgraphs from one thread are passed to the next thread `mod NUM_THREADS`, and handled asynchronously. In the place of L, each thread maintains a sorted list of subgraphs it has passed to the next thread, which is cleared when the order of subgraphs being handled by that thread is incremented.

2.2 Complexity

Since the brute-force search space of the algorithm is confined to the actual dense subgraphs of G, the worst-case performance is when graph G is a clique. A clique with n vertices has 2^n possible combinations of vertices, each combination forming a fully connected subgraph, resulting in a complexity of $O(2^n)$. When a dense subgraph is not a clique, the worst-case is ameliorated by the ordering property of the theorem, which forces initial seed graphs to be of high density, e.g. searching for subgraphs with density at least 0.7 forces all initial seed graphs with vertices to be a clique, and subsequent 4-vertex seed graphs to have 5 of 6 possible edges, etc. This works well for networks that exhibit a power-law distribution of vertex degree, of which biological networks are one example.

2.3 Implementation

ODES is implemented with pthreads in the **C** programming language as a function call, returning a list of maximal dense subgraphs in adjacency list format. The function signature consists of the input graph in adjacency list format, the minimum desired density of a subgraph, and the minimum number of vertices that a dense subgraph must contain.

2.4 Performance

Fig. 2.2 records timings using 4 threads on graphs G_i of 20 vertices each, whose density varies over values that range from a minimum cutoff density D to 1, and where G_i is not embedded in any larger graph. Six plots are shown, corresponding to each of six values for D, and the time shown is how long it takes for the algorithm to return G_i .

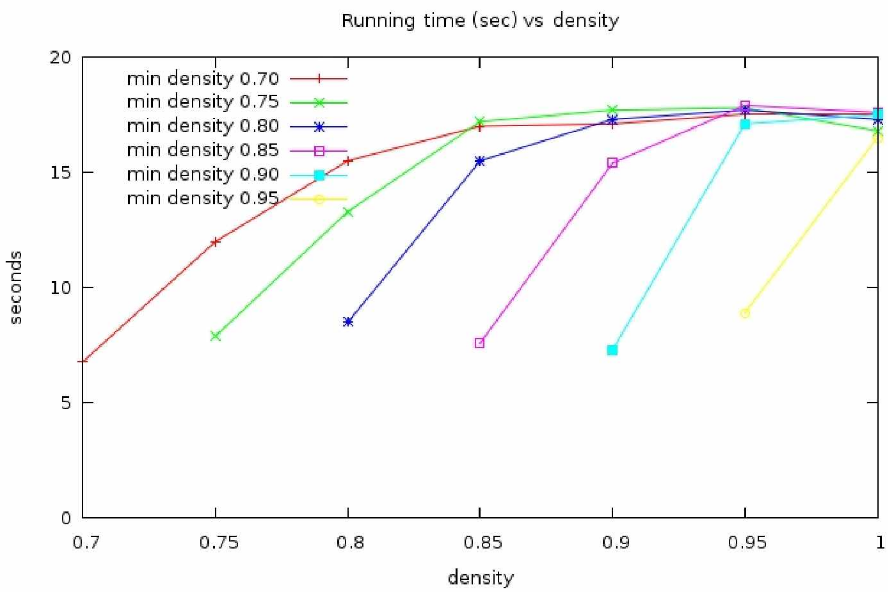


Fig. 2.2: Running time vs density

We observe that the ordering property of the theorem has the greatest effect on reducing the search time when the actual density of G_i is close to the cutoff D, rather than being appreciably denser than D, showing a three-fold speedup for densities close to D, and exhibiting roughly constant running time for densities of G_i close to D as D increases.

Embedding any G_i within a 10,000-vertex graph, where the average degree of a vertex is 4, increases the run time approximately 10 percent.

2.5 Enhancements

Fig. 2.2 suggests that as the number of threads available on modern multi-core processors increases, the algorithm should be restructured to run over a collection of density bins, where a set of threads is dedicated to each density bin that handles only those subgraphs whose density has decreased below the cutoff for the bin above it, thus finding all maximal dense subgraphs of the same order in roughly constant running time for densities not close to 1.0. All single-edge subgraphs would start off in the bin that handles the highest density range.

Profiling of the running code reveals a large percentage of time spent in binary searches of the sorted list maintained by each thread of subgraphs passed to the next thread, suggesting that a suitable hash function for graphs may increase performance.

Finally, an indication of which graphs overlap, and where, is desirable.

3: Pipeline Architecture

3.1 Background

Techniques from algorithms that deconvolve biological networks from microarray data [Faith et al., 2007, Hu et al., 2005, Margolin et al., 2006, Zhou et al., 2005] will be employed to determine expression correlation between two genes. Deconvolving a biological network from microarray data begins with expression correlation, but goes one step further by using correlation data with additional assumptions to infer the biological network responsible for generating the data. Graphically, a declaration of expression correlation between two genes manifests as an undirected edge between the two genes in a graph, while the deconvolution of a biological network from microarray data manifests as (possibly directed) edges from transcription factors to the genes they regulate.

Biological network deconvolution is a significantly harder problem than expression correlation determination, but depends on accurate correlation methods nonetheless. One problem is high false-positive rates that infer a direct connection between two proteins, when in fact the interaction is the result of some intermediate or cascade [van Someren et al., 2002]. Another is the dimensionality problem, where a system with tens of thousands of genes is sampled in a typical experiment by fewer than 20 microarrays [van Someren et al., 2001]. Deconvolution algorithms thus require a large amount of microarray data; such data is typically collected from different experimenters under a variety of conditions, which presents another problem since the data is generally from diverse platforms, where expression values are not directly comparable. Noise in the data compounds all of the problems.

In addition, there are not enough known networks against which to test deconvolution algorithms. This dissertation focuses only on the correlation determination problem by integrating and extending the work of others that have addressed one or more of the deconvolution problems enumerated.

As a side note, deconvolution algorithms have typically been applied to one organism under a variety of conditions, however, there is no *a priori* reason that all the data be from the same organism, as long as there is sufficient overlap in mapping genes in one organism to genes in another, and some way is made for comparison of expression values across microarray platforms. As the amount of microarray data increases, the prospect of cross-species data integration offers the opportunity to explore conservation of important networks [Zhou & Gibson, 2004]. Similarly, conserved modules across species may also be explored.

Techniques from three algorithms are to be integrated in this work: the Mining Coherent Dense Subgraphs (CODENSE) algorithm [Hu et al., 2005], the Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE) [Margolin et al., 2006], and the Context Likelihood of Relatedness (CLR) algorithm [Faith et al., 2007]. Their integration will be accomplished by augmenting the simple Pearson's correlation measures contained in the CODENSE algorithm with the information-theoretic mutual information methods of non-linear correlation contained in the ARACNE and CLR algorithms, and the Z-score method used in CLR. This modified CODENSE method will be subject to a sensitivity analysis on a suite of synthetic microarray data generated by a Random Network Generator (RANGE), developed by the author [Long & Roth, 2008].

An overview is given next of the three algorithms whose key ideas will be integrated into a single algorithm.

3.2 Algorithm for the Reconstruction of Accurate Cellular Networks

The Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE) [Margolin et al., 2006] is an information theoretic reverse engineering algorithm to identify direct transcriptional interactions in mammalian cellular networks.

3.2.1 Information Theory

C. E. Shannon developed Information Theory in the mid 20th century. The theory originally applied to communication networks, where the problem of determining the maximum information transmission capability in the presence of noise was considered. Given some source of information, a receiver, and a communication channel connecting the two, the channel may introduce noise into the signal, changing what is received from what is sent. It is thus desirable to encode sent messages in such a way that errors may be detected and corrected, while transmission rate is maximized. Here we describe only those aspects of the problem that relate to measuring the degree of correlation between input and output; for microarray data it is the correlation between two expression levels.

3.2.2 Entropy

Entropy is a measure of the amount of information contained in some symbol or message, or alternatively, a measure of the uncertainty about what symbol or message a source may transmit. Assume we are to send one of n messages, denoted $x_i, 0 \leq i \leq n-1$. If the probability of sending message x_i is $p(x_0)=1$ and $p(x_i)=0$ for $i>0$, then x_0 is always chosen, and our uncertainty about what will be sent is zero, i.e. we get no additional information from the source when the message is received. On the other hand, if $p(x_i)=\frac{1}{n}$, for n atomic events, then our uncertainty about what will be sent is maximal, as is the increase in the amount of information we get from receiving what gets sent. The self-information in a message is defined as

$$I_i = -\log_2 p(x_i) \qquad \textit{Eqn. 3.1}$$

and may be thought of as the minimum number of bits necessary to encode the information about which symbol or message is sent. For the case of flipping a fair coin, $p(x_i)=\frac{1}{2}, i=1,2$ and $I_i=-\log_2\frac{1}{2}=1$ bit of information. The expected value of the self-information is known as the entropy

$$H(X) = -\sum p(x_i)\log_2 p(x_i) \qquad \textit{Eqn. 3.2}$$

where X is a random variable defined over a sample space E of atomic events x_i so $\sum_i p(x_i)=1$. For the coin flipping example, the range of entropy for different probabilities of a coin is plotted in Fig. 3.1.

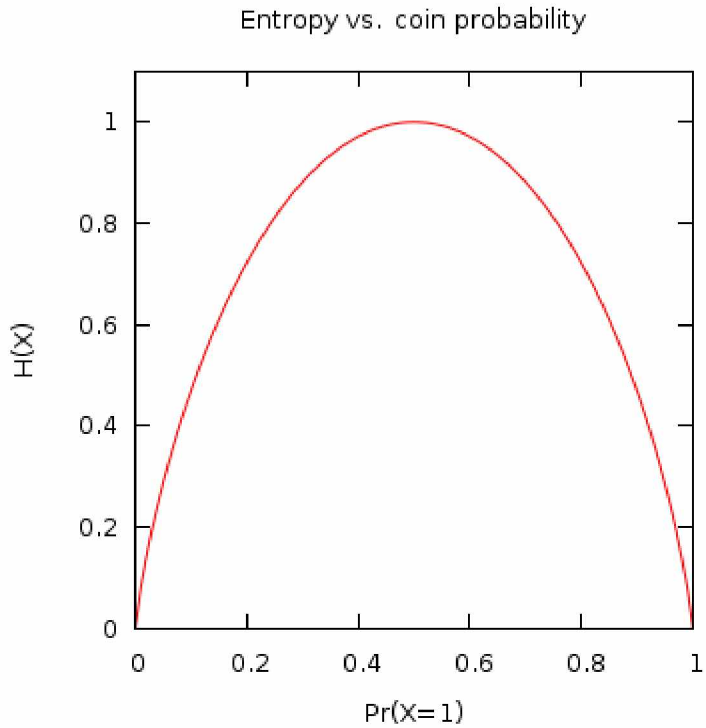


Fig. 3.1: Entropy vs coin probability

Given a second random variable Y defined over another sample space F of atomic events y_j, we can form the bivariate product space $E \times F$ and define

$$H(Y) = -\sum_i p(y_j) \log_2 p(y_j) \ ,$$

along with the joint entropy for the bivariate system

$$H(X,Y) = -\sum_i \sum_i p(x_i, y_j) \log_2 p(x_i, y_j) \qquad \textit{Eqn. 3.3,}$$

where $p(x_i, y_j)$ is the probability of x_i and y_j occurring together.

3.2.3 Mutual Information

A discrete memoryless channel connecting source to receiver is characterized by a matrix with entries $a_{ij} = p(y_j \mid x_i)$. Known as the channel matrix, $\prod$ quantifies the probability of receiving y_j if x_i is transmitted. In a system with no noise, the channel matrix can be written as a diagonal matrix, i.e. a unique y_j is received for each x_i . It is memoryless because the a_{ij} are do not depend on previous transmissions. Channel matrices with memory are not considered here.

Given an input distribution of $X, p(x)$ to a channel $\prod$, the output distribution for *Y* is given by

$$P(Y = y_j) = p(y_j) = \sum_i p(x_i) p(y_j \mid x_i), \quad j = 0,1, \ldots, n-1 \ .$$

This may be viewed as a channel acting on an input to produce an output, and we define the information processed by the channel as

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \qquad \textit{Eqn. 3.4}$$

also know as the "mutual information" or "transinformation", a measure of the correlation between *X* and *Y*.

For microarray data, *X* and *Y* are expression levels of transcription products, and the mutual information $I(X;Y)$ represents the degree of correlation between the two, being zero precisely when *X* and *Y* are independent of each other. Note that mutual information captures non-linear as well as linear correlation between two variables.

In ARACNE, two genes are declared related if their mutual information score is above a specified cutoff value, and the association is not an indirect one, determined by the application of the data processing inequality [Cover & Thomas, 2006], thus addressing the high false-positive rate among deconvolution algorithms. ARACNE is provably correct for asymptotically exact network reconstruction if loops in the network have negligible effect, and works well even in complex topologies with numerous loops. Source code is available for ARACNE.

3.3 Context Likelihood of Relatedness

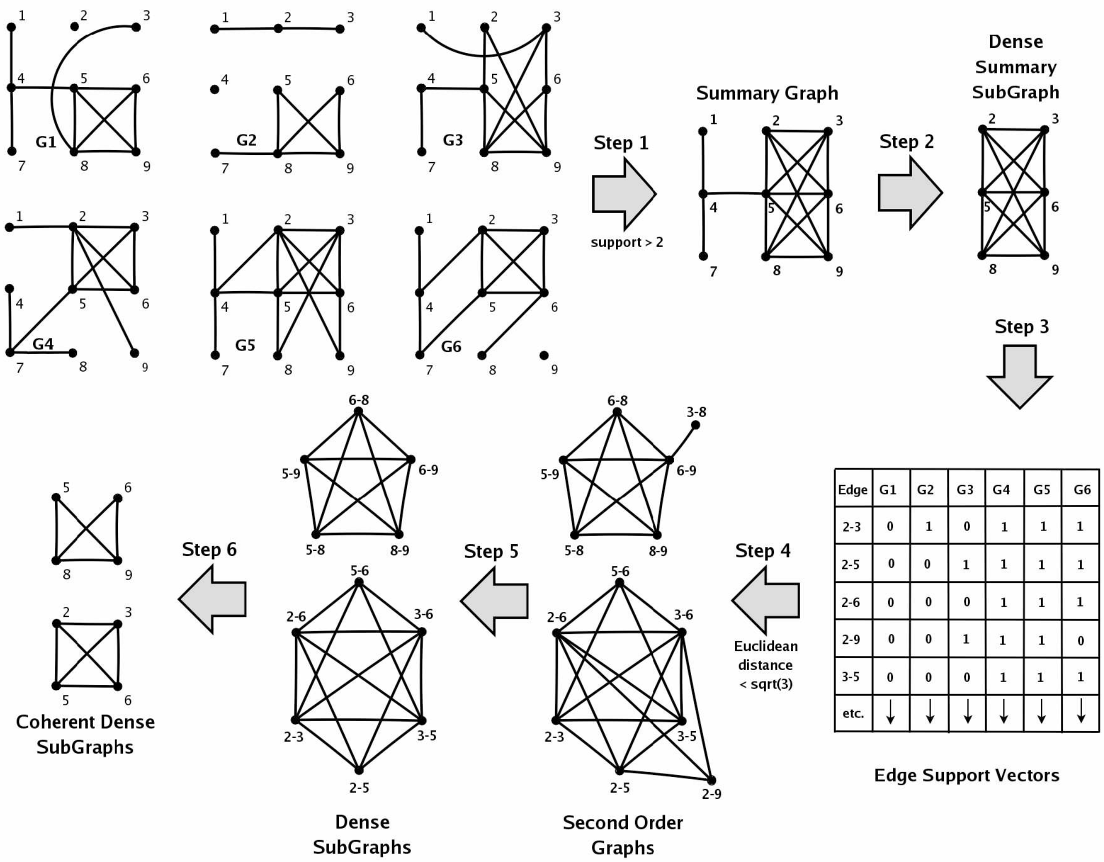
The Context Likelihood of Relatedness (CLR) algorithm [Faith et al., 2007] is an ARACNE-like algorithm designed to increase the contrast between direct and indirect interactions by examining the network context of each relationship determined by either a mutual information or Pearson's correlation score. Network context determines an adaptive correction step based on computation of the background distribution of mutual information scores (or Pearson's correlation) for the context. Direct interactions are those whose mutual information score is significantly above the background as measured by the Z-score, or normalized score. Source code is available for CLR.

3.4 CODENSE

Given a collection of microarray data sets generated from potentially heterogeneous platforms, where each set represents expression under a given condition, the Mining Coherent Dense Subgraphs (CODENSE) algorithm [Hu et al., 2005, Zhou et al., 2005] looks for network modules that are conserved across the collection. It first extracts expression patterns from each set (first-order analysis) by categorizing all gene pairs by Pearson's correlation, and then examines correlated occurrences of the expression patterns across the collection of data sets (second-order analysis), using a linear correlation method such as Euclidean distance or Pearson's correlation. First-order patterns that are highly correlated across the collection of data sets represent genes in modules likely to be functionally related. These highly correlated first-order network patterns are conserved biological modules, and are the object of this dissertation. Since correlation is among expression patterns across a collection of data sets, CODENSE has the advantage that the data sets need not be from the same microarray platform, and in the case of meta-genes, the data need not even be from the same species. Source code is unavailable for CODENSE, but a binary with documentation is available from the Zhou Lab (http://zhoulab.usc.edu/CODENSE/).

3.5 The Enhanced CODENSE Pipeline

Fig. 3.2 depicts the original CODENSE pipeline overview, which remains unchanged after modification by the information theoretic and context likelihood methods employed by ARACNE and CLR.



Edge	G1	G2	G3	G4	G5	G6
2-3	0	1	0	1	1	1
2-5	0	0	1	1	1	1
2-6	0	0	0	1	1	1
2-9	0	0	1	1	1	0
3-5	0	0	0	1	1	1
etc.	↓	↓	↓	↓	↓	↓

Fig. 3.2: The CODENSE pipeline

Each pipeline stage is outlined next, including modifications. Chapter 4 covers the software implementation of each stage.

3.5.1 Step 0: First Order Graphs

Generate a first-order pattern graph for each collection of microarray data, where each such collection represents one experiment, composed typically of a series of microarray slides we'll call *snapshots*. Whether from one individual organism, perhaps a time series for that organism, or from a group of organisms, the snapshots in an experiment are usually associated by being representations of expression levels during a given state. Each graph is generated by examining each pair of expression levels for correlation across the collection. For CODENSE, an edge is inferred between two genes when the expression correlation is above a Pearson's correlation cutoff score. The enhanced CODENSE pipeline adds the option of using the information theoretic and context likelihood methods employed by ARACNE and CLR to infer these edges. Fig. 3.2 shows first-order pattern graphs from 6 experiments, each composed of a series of snapshots.

3.5.2 Step 1: Summary Graph

Produce, from the collection of graphs, a summary graph consisting of those edges that appear in at least X of the graphs, where X is called the *support*, which is $\geqslant 3$ for this example. In the enhanced pipeline implementation, *support* is redefined as the fraction of a collection that an edge must appear in to be included in the summary graph.

3.5.3 Step 2: Summary Graph Dense Subgraphs

Find the dense subgraphs in the summary graph, where dense means that some high fraction of the total number of possible edges is present in the subgraph. Density is defined as $d=\frac{2e}{n(n-1)}$, where e is the number of edges, and n is the number of nodes, or vertices. Dense subgraphs in the summary graph are a superset of dense first-order subgraphs that occur sufficiently often across the collection of first-order graphs, but not every dense subgraph in the summary graph is necessarily a dense subgraph in a first-order graph.

3.5.4 Step 3: Edge Support Matrix

Build an edge support matrix from the dense summary graph. Matrix entries are indexed first by an individual dense summary graph edge, and second by the first-order graphs in which the particular edge appears. In Fig. 3.2, the matrix entries are simple boolean values, but could also be edge weights. Boolean values are used in this study.

3.5.5 Step 4: Second Order Graphs

Use the edge support matrix to construct second-order graphs, where each vertex in a second-order graph represents an edge in the dense summary subgraph, and an edge in a second-order graph represents a correlation of occurrence between a pair of edges in the dense summary subgraph across the first-order graphs, where correlation is measured by an euclidean distance similarity score.

3.5.6 Step 5: Second Order Dense Subgraphs

Find the dense subgraphs in the second-order graphs. Dense subgraphs in the second-order graphs represent groups of edges in the dense summary subgraph that mostly occur together across the first-order graphs.

3.5.7 Step 6: Coherent Dense Subgraphs

Convert each dense second-order subgraph from Step 5 into a first-order graph, find the dense subgraphs in those first-order graphs, and output them. Coherent dense subgraphs are those dense subgraphs in the first-order graphs whose edges exhibit high correlation across the whole collection of graphs (second-order expression correlation).

In the original CODENSE pipeline, graph mining for subgraphs in steps 2, 5, and 6 is accomplished by an heuristic algorithm [Hu et al., 2005] known as MODES (Mining Overlapping Dense Subgraphs). In the enhanced pipeline, MODES is replaced by ODES, the exact method described in chapter 2.

4 Pipeline Implementation

4.1 CodenseMI

`CodenseMI()` calls the modules that implement each pipeline step of chapter 3. It takes one argument, a file named 'fileList' that contains a list of microarray data set file names that conform to the format specified in `ReadMicroarrayData()`, described later. It also reads a 'parameters' file, Table 4.1, that overrides `CodenseMI()` defaults.

Table 4.1: Pipeline parameters file

```
#define DO_MI               1
#define DO_Z                1
#define CO_DENSITY          0.761928
#define SG_DENSITY          0.712562
#define SO_DENSITY          0.828075
#define GENE_THRESHOLD      0.000613
#define PC_CUTOFF           0.700951
#define SOG_CUTOFF          0.512643
#define SUPPORT             0.223568
#define PLUS_CORREL         1
#define COMINNODES          4
#define SGMINNODES          4
#define SOMINNODES          4
#define ZPC_SLICE           0.00013298
#define ZMI_SLICE           0.00027976
#define DIM                 256
#define MI_CUTOFF           4.244286
#define NEG_CUTOFF          -0.357853
#define PEAK                8192.000000
```

Descriptions of pipeline parameters file members

`DO_MI`	- flag to use Mutual Information (MI) in first-order graph edge determinations.
`DO_Z`	- flag to use Z-scores in first-order graph edge determinations.
`CO_DENSITY`	- graph density cutoff score for coherent dense subgraphs.
`SG_DENSITY`	- graph density cutoff score for summary graph dense subgraphs.
`SO_DENSITY`	- graph density cutoff score for second-order graph dense subgraphs.
`GENE_THRESHOLD`	- fraction of expression sum that a gene must exceed for a correlation calculation to proceed.
`PC_CUTOFF`	- minimum PC score for an edge to join a first-order graph
`SOG_CUTOFF`	- minimum similarity score for an edge to be included in a second-order graph
`SUPPORT`	- fraction of first-order graphs an edge must be a member of, to be included in the summary graph
`PLUS_CORREL`	- always set to 'true', only positive PC scores are considered
`COMINNODES`	- minimum number of vertices that must exist in a dense coherent subgraph
`SGMINNODES`	- minimum number of vertices that must exist in a dense summary subgraph
`SOMINNODES`	- minimum number of vertices that must exist in a dense second-order subgraph
`ZPC_SLICE`	- Z-score cutoff for PC
`ZMI_SLICE`	- Z-score cutoff for MI
`DIM`	- joint probability matrix dimension, $mod\, 32 = 0$
`MI_CUTOFF`	- minimum MI score for an edge to join a first-order graph
`NEG_CUTOFF`	- no strong negative PC should infer an edge, even if MI is high, so PC must be $\geqslant$ this value
`PEAK`	- central maximum of the kernel density function used in MI calculations

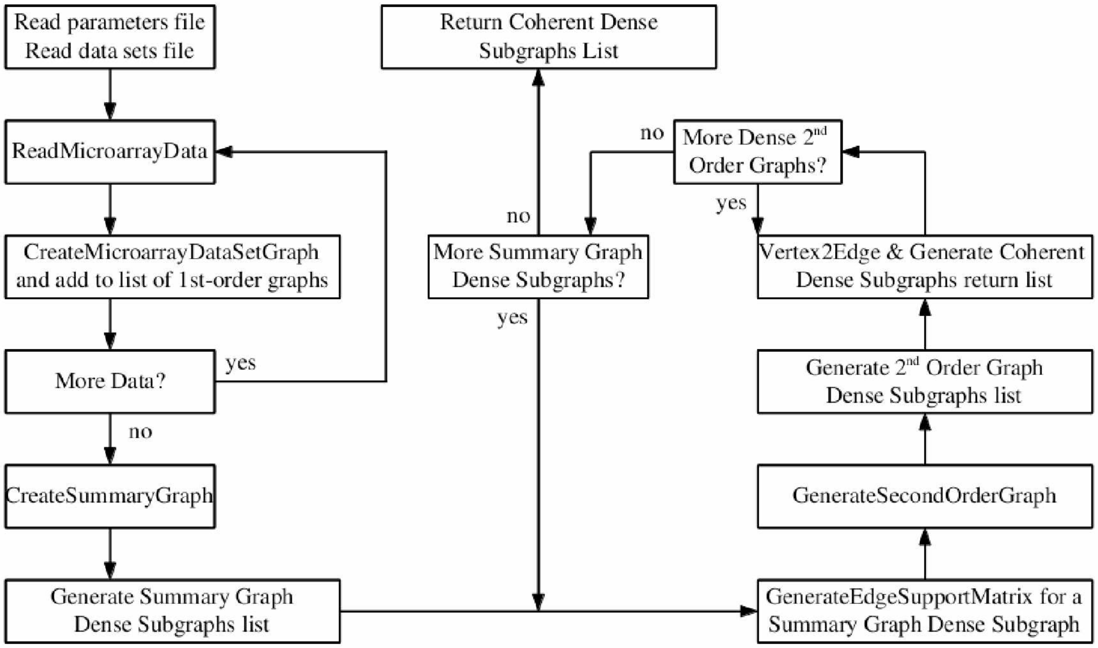


Fig. 4.1: `CodenseMI()` *flowchart*

A flowchart of how `CodenseMI()` implements the module discovery pipeline is presented in Fig. 4.1, followed by a summary of the pipeline steps of Fig. 3.2 using the routines that are called by `CodenseMI()`:

1) `ReadMicroarrayData()` and `CreateMicroarrayDataSetGraph()`
2) `CreateSummaryGraph()` operating on the collection of graphs from step 1
3) `ODES()` (Overlapping DEnse Subgraphs) operating on the summary graph
4) `GenerateEdgeSupportMatrix()` operating on the dense summary graph
5) `GenerateSecondOrderGraph()` operating on the edge support matrix
6) `ODES()` (Overlapping DEnse Subgraphs) operating on the second-order graphs
7) `Vertex2Edge()` operating on each dense second-order graph, followed by `ODES()` (Overlapping DEnse Subgraphs) operating on the resulting first-order graphs

`CodenseMI()` returns a list of `gnGraph` pointers, where each `gnGraph` represents a dense subgraph in adjacency list format. This format is implemented as a list of `graphNode` structures, each containing a vertex with associated edges and edge weights:

```c
typedef struct graphNode_jlong
{
  char *label;    /* from microarrayDataSet */
  char **edges;   /* array of graphNode labels the node connects to */
  double *edgeWeights;
  int numEdges;
} graphNode;

typedef struct gnGraph_jlong
{
  graphNode **gnList;
  int numNodes;
} gnGraph;
```

Each of the pipeline routines called by `CodenseMI()` are now discussed in detail.

4.2 ReadMicroarrayData

`ReadMicroarrayData()` reads a text file with a header followed by data, viz

```
probeID      Array1Dye1  Array2Dye1  Array3Dye1  Array4Dye1    ...
1415670_at   9.172966    9.236673    8.789873    8.698874      ...
1415671_at   10.665967   10.677811   10.463303   10.464032     ...
    .            .           .           .           .          .
    .            .           .           .           .            .
    .            .           .           .           .              .
```

The first column is a probe identifier representing the target for a specific **mRNA**, and subsequent column values in the same row for a probe ID are the expression values for the targeted **mRNA** from different snapshots, that is, data from either a series of different microarray slides or data as determined by a series of Next Generation Sequencing (NGS) procedures. The data is returned as a pointer to a *microarrayDataSet* struct:

```
typedef struct microarrayDataSet_jlong
{
  int numCols, numRows;
  char **labels; /* each label is unique */
  double *data;  /* really a data table  */
} microarrayDataSet;
```

The probeIDs are the *labels*, and the 2D array of expression values is the data, with *numCols* columns, and *numRows* rows.

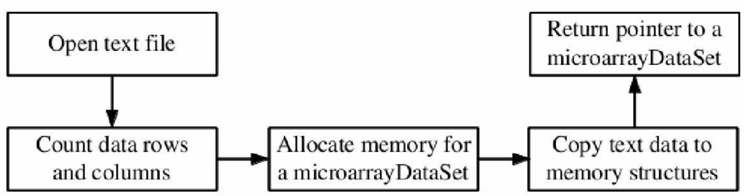


Fig. 4.2: `readMicroarrayData()` *flowchart*

4.3 CreateMicroarrayDataSetGraph

`CreateMicroarrayDataSetGraph()` generates a first order graph for a microarray data set. In the original CODENSE pipeline, an edge is declared between two genes if their expression correlation as determined by Pearson's correlation is above some cutoff value, Fig. 4.3.

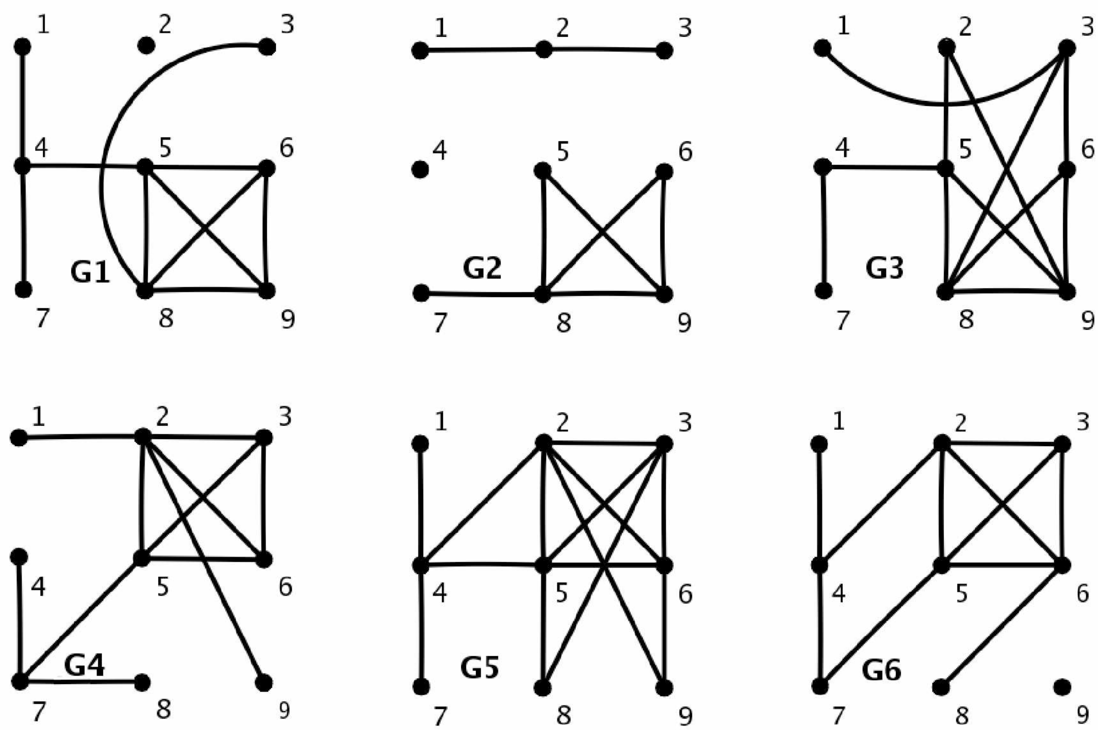


Fig. 4.3: Six microarray data set graphs

For the modified CODENSE pipeline, an edge between two genes may be similarly declared, with two additional options available to either augment or replace the declaration. Application of the Z-score method as used in the CLR algorithm augments a high Pearson's correlation by imposition of the additional stipulation that the correlation between two genes also be significant. The CLR algorithm first calculates the distribution of correlations in the population, and then uses a cutoff for the standard score, or Z-score, for each gene's correlation value. As used in the enhanced pipeline, this stipulation limits the number of false-positive edge declarations, because in addition to being significant, the correlation must also remain high. Using only the Z-score method without requiring that the correlation be high would admit the possibility of significant correlations across a low scoring distribution being declared edges.

The second option available invokes a correlation based on the information-theoretic mutual information score, calculated only when the Pearson's correlation method fails to declare an edge. The Z-score method may also be invoked, in the same manner as previously, requiring that a mutual information score be both high, and significant. See Fig. 4.4 for the `CreateMicroarrayDataSetGraph()` flowchart.

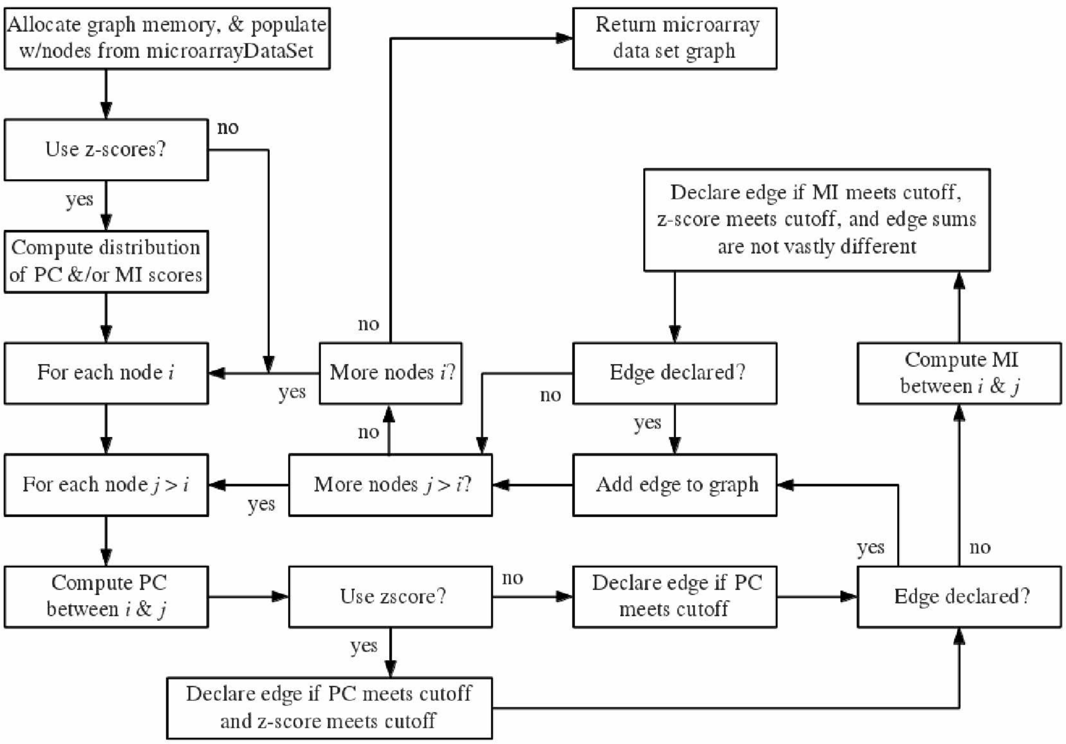


Fig. 4.4: `CreateMicroarrayDataSetGraph()` *flowchart*

For both options, we wish to eliminate consideration of very low expression values in the data that may be considered noise. No edge is declared between two genes if one of the following conditions holds:

1) the sum of one gene's expression values across a data set is less than some small fraction of the other gene's expression values.
2) gene expression sums for both genes are less than 10^{-12} femtoliters.
3) more than half of the expression values for one gene in a data set are less than 10^{-13} femtoliters.

For criteria 1), the fraction in question is a parameter of the pipeline, known as the 'gene threshold', GT.

4.4 CreateSummaryGraph

`CreateSummaryGraph()` ingests a list of microarray data set graphs, each member of the list created from a call to `CreateMicroarrayDataSetGraph()`. It then sorts all the vertices from the collection of microarray data set graphs (the `graphSet`), removing duplicates. The summary graph is initially populated with these vertices, and edges are added to the summary graph only if they occur enough times across the `graphSet`, i.e. the sum of the edge weights for an edge exceeds

$$threshold = support * numberOfDataSets \ .$$

So for every unique vertex label in the summary graph:

1) look for that label in every graph of `graphSet`.
2) collect all edges attached to it in every graph.
3) sum the edge weights for each edge.
4) if the sum meets or exceeds the *threshold*, add the edge to the summary graph.

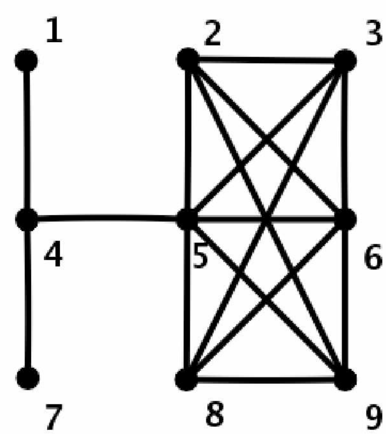


Fig. 4.5: Summary Graph constructed from graphs in Fig. 4.3

Once the summary graph has been constructed, the Overlapping Dense Subgraph Algorithm extracts qualifying dense subgraphs from the summary graph that exceed a specified cutoff density $\geqslant 0.5$, see Fig. 4.6.

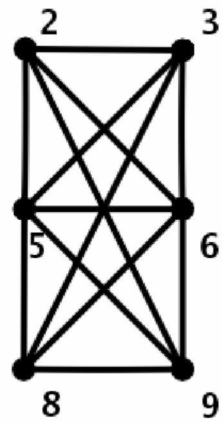


Fig. 4.6: The Dense Summary Subgraph obtained from Fig. 4.5

These dense subgraphs are a superset of dense first-order subgraphs that occur a sufficient number of times across the first-order graph collection. Note, however, that not every dense subgraph in the summary graph is necessarily a dense subgraph in a first-order graph. In this case, the dense summary subgraph consists of edges that occur a significant number of times across different first-order graphs.

4.5 GenerateEdgeSupportMatrix

`GenerateEdgeSupportMatrix()` returns an e*dge support matrix*, a table of edges from the summary graph, versus the first-order graphs in which they appear, see Fig. 4.7. It is represented as a list of structures, each of which contains the name of an edge in a dense summary subgraph, and a vector for that edge enumerating its weight in each first-order graph.

Edge weights in the enhanced pipeline are boolean, either 1.0 or 0.0, for either present or absent, although the code can handle arbitrary floating point edge weights.

An edge support matrix has the same basic format as a microarray data set, where probes are replaced by edges, and expression values are replaced by edge weights. This fact will be leveraged in the next section, where generation of second-order graphs is discussed.

Edge	G1	G2	G3	G4	G5	G6
2-3	0	1	0	1	1	1
2-5	0	0	1	1	1	1
2-6	0	0	0	1	1	1
2-9	0	0	1	1	1	0
3-5	0	0	0	1	1	1
etc.	↓	↓	↓	↓	↓	↓

Fig. 4.7: The Edge Support Matrix for Fig. 4.6

4.6 GenerateSecondOrderGraph

The vertices in second-order graphs are edges in first-order graphs, and an edge between two vertices in a second-order graph represents correlation of occurrence between two first-order edges across the collection of first-order graphs. `GenerateSecondOrderGraph()` produces such graphs from an edge support matrix, see Fig. 4.8.

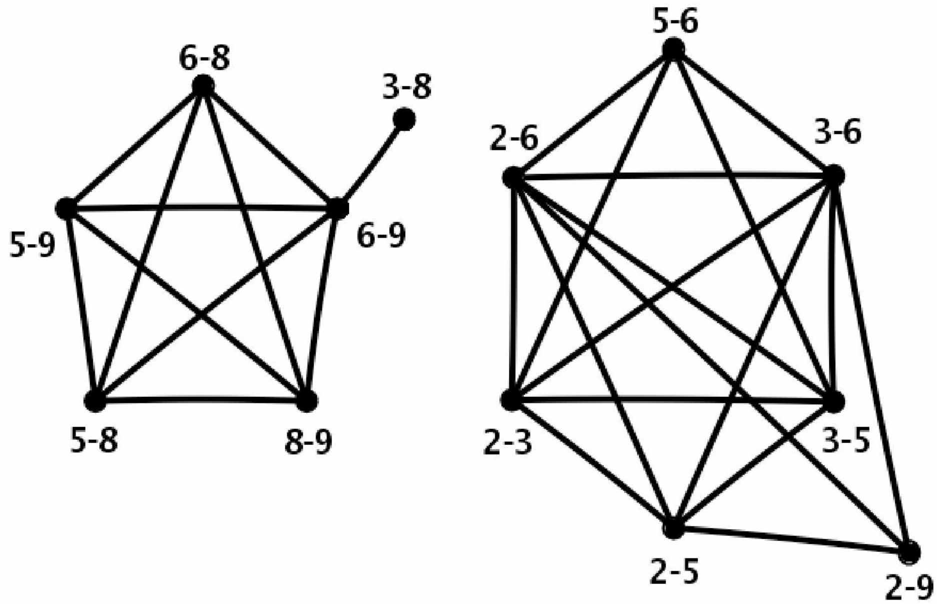


Fig. 4.8: Second-order Graphs constructed from Fig. 4.7 Edge Support Matrix

Second-order graphs are constructed from the edge support matrix in almost the same manner that a first-order graph is constructed from a microarray data set, if probes are replaced by edges, and expression values are replaced by edge weights. Generating a second-order graph can then be reduced to transforming the edge support matrix into a microarray data set, and calling `CreateMicroarrayDataSetGraph()`, with the caveat that the correlation be measured by an euclidean distance similarity score, rather than a Pearson's or Mutual Information score. Given that edge support matrix values are either `1` or `0`, the similarity score between two rows in this case is simply the fraction of array values in one row that are the same as the values in the other row. Once the second-order graphs are generated, their dense subgraphs are extracted, see Fig. 4.9. Dense subgraphs in the second-order graphs represent groups of edges in the dense summary subgraph that mostly occur together across the first-order graphs.

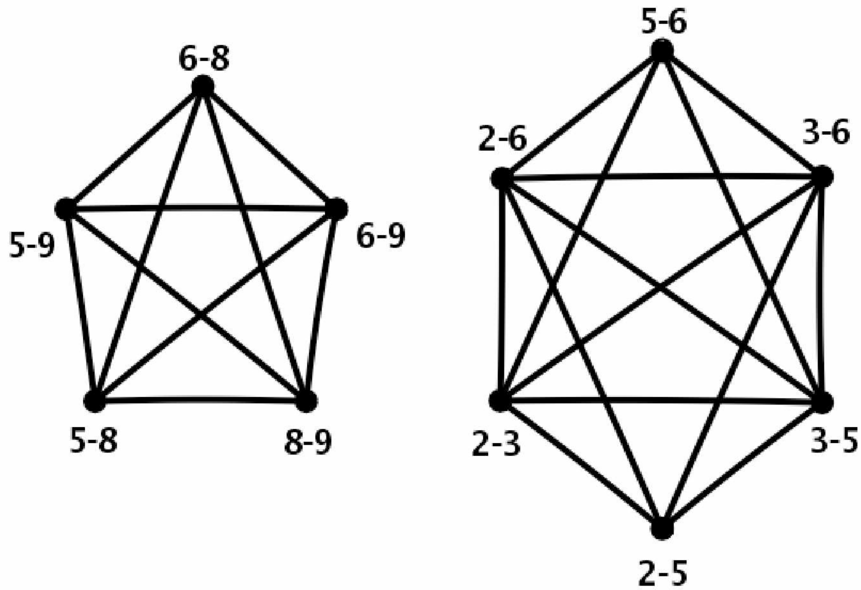


Fig. 4.9: Dense Second-order Subgraphs of Fig. 4.8

4.7 Vertex2Edge

`Vertex2Edge()` converts each dense second-order subgraph into a first-order graph. Any dense second-order subgraph D is a graph whose vertices are labeled "`L1<-->L2`", where $L1$ and $L2$ are vertices in a graph G that are adjacent. We call G the Vertex-to-Edge graph of D.

In CODENSE, a coherent dense subgraph is a dense subgraph in some Vertex-to-Edge graph G of some dense subgraph D in a second-order subgraph of the Summary Graph.

Restated, coherent dense subgraphs are those dense subgraphs in the first-order graphs whose edges exhibit high correlation across the whole collection of graphs (second-order expression correlation). Fig. 4.10 shows the graphs of Fig. 4.9 converted to first-order graphs, coincidently their own dense subgraphs, and hence the coherent dense subgraphs of Fig. 4.9.

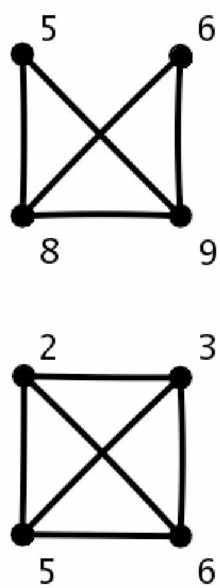


Fig. 4.10: Coherent Dense Subgraphs of Fig. 4.9

5 Regionalized Sensitivity Analysis

The "regionalized sensitivity analysis" (RSA) method has been shown to be useful for the problem of selecting parameter values in a model for which only sparse data is available [Hornberger & Cosby, 1985]. The RSA method determines the most important parameters in a model for simulating a desired behavior, and then a statistical optimization across only those important parameters can be performed to pick the best values for the model. In this study, the RSA method is used only to determine what parameters are most important for simulating a desired behavior. Quantification of the impact of the most important parameters on model output may suggest areas of the model that need further refinement.

Since precise values for many parameters are typically unknown, an estimated uniform probability density function (drawn from the literature where possible) for each parameter in a model is supplied to the RSA method, from which individual values for Monte Carlo runs are sampled. At the end of each run, the output of the model is examined by a binary objective function for either conformance or non-conformance to one or more predetermined behavioral criteria, and a cumulative distribution for both the conforming and non-conforming behavior for each parameter is updated.

A Kolmogorov-Smirnov two-sample test (K-S two-test) is used to compare the two behaviors, where the K-S statistic is defined as the maximum separation $d_{c,n}$ between the empirical distributions for the conforming and non-conforming behaviors given by

$$d_{c,n} = \sup|S_c(x) - S_n(x)| \qquad \textit{Eqn. 5.1}$$

where $S_c(x)$ and $S_n(x)$ are the sample (empirical) distributions for the conforming and non-conforming behaviors respectively.

Fig. 5.1 illustrates these cumulative distributions, along with the parent *a priori* cumulative distribution for the uniformly distributed parameter in question.

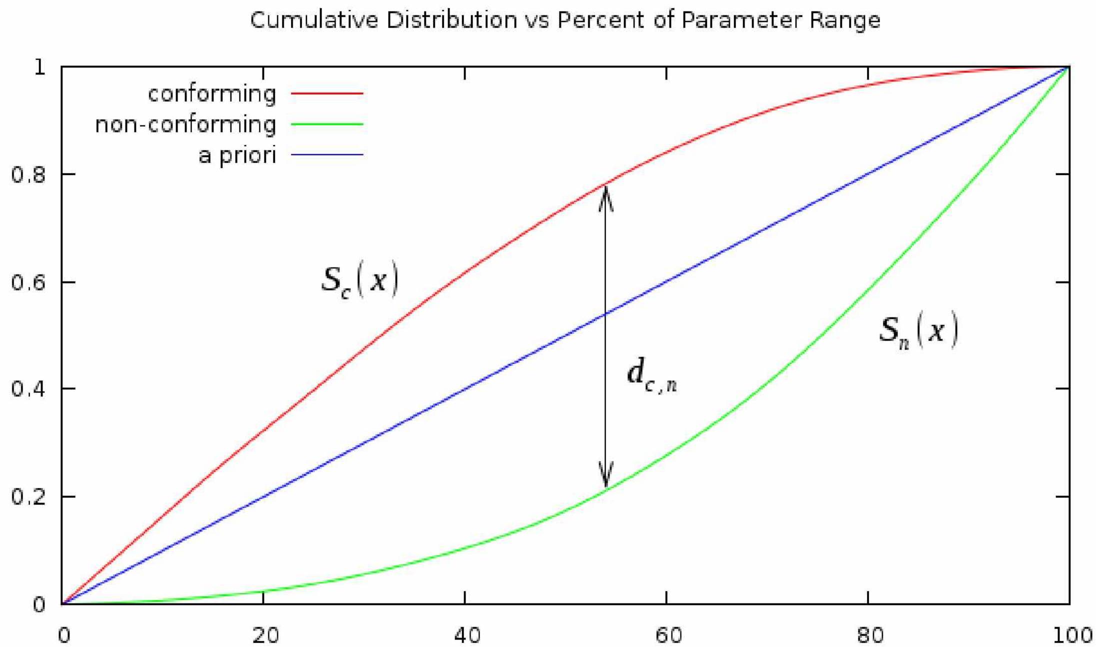


Fig. 5.1: Cumulative distributions

Since the K-S two-test is sensitive to differences in the shape of the empirical cumulative distributions, $d_{c,n}$ serves as a measure of how important a parameter is in a model with respect to the chosen objective function. Large values of $d_{c,n}$ for a parameter indicate that the underlying probability distributions for conformance and non-conformance differ, and thus that parameter influences model output more than one whose conformance and non-conformance distributions are closer to the parent.

5.1 Application

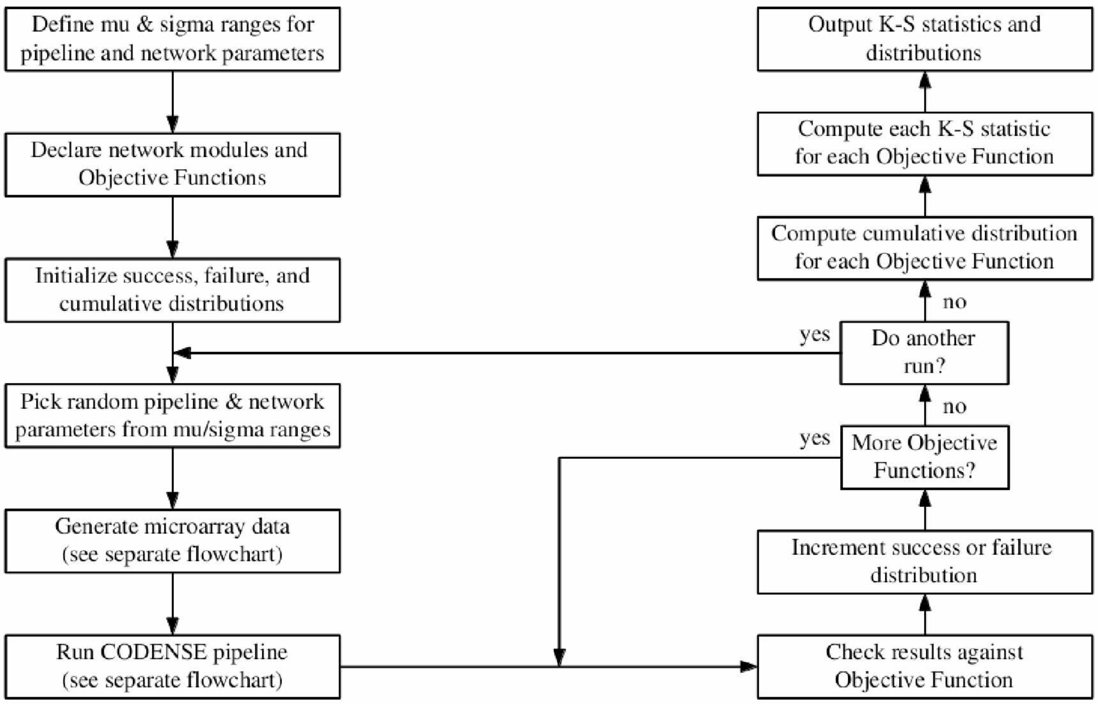


Fig. 5.2: RSA driver flowchart

To apply the RSA method to the Coherent Dense Subgraphs pipeline, a uniform distribution range for each model parameter must be chosen, along with one or more objective functions. Model parameter ranges are chosen from the literature where possible, or estimated if no guidelines exist. Two sets of model parameters exist, one for pipeline parameters, and another for synthetic network parameters, delineated in the next two sections. Fig 5.2 shows the flowchart for the RSA driver, coded as `sensitivitySIM.c`.

5.1.1 Pipeline Parameters

`DCO` - minimum density for a dense subgraph in coherent dense subgraphs

`DSG` - minimum density for a dense subgraph in summary graph dense subgraphs

`DSO` - minimum density for a dense subgraph in second order dense subgraphs

`GT` - gene threshold: fraction of expression sum for one gene that the other gene has to exceed for a correlation calculation to proceed

`PC` - PC score that an edge must meet or exceed:
for simple Pearson's, values >= PC are an edge
for DOZ, values >= PC and in top Z-score slice are an edge

`SOGCO` - Similarity score that an edge must meet or exceed in GenerateSecondOrderGraph() to be a 2nd-order graph edge

`SUPP` - an edge must exist in at least 'support' of the microarray data sets to be part of the summary graph

Flags:

`DOMI` - mutual information flag

`DOZ` - Z-score flag, used for PC & MI

Mutual Information Parameters, used when `DOMI = 1`:

`DIM` - joint probability matrix dimension, mod 32 = 0

`MI` - MI score that an edge must meet or exceed:
for simple mutual information, values >= MI are an edge
for DOZ, values >= MI and in top Z-score slice are an edge

`NEG` - 1) if using Mutual Information and only positive Pearson's Correlation to infer an edge, it must be above this value, i.e. no strong negative Pearson's should infer an edge, even if MI is high (the usual mode)

2) if using negative Pearson's Correlation to infer an edge, it must be below this value (not used, future experimental feature only)

`PEAK` - central max of kernel density function

`SFACT` - factor by which the sum of one gene's expression levels must be less than the other for an MI calculation to proceed

Z-score Parameters, used when `DOZ = 1`:

`ZMI` - Z-score cutoff for MI, DOMI must also be set

`ZPC` - Z-score cutoff for PC

5.1.2 Synthetic Network Parameters

`IDS` - number of snapshots in a microarray data set

`MDS` - number of microarray data sets

`MUB` - mu for normal dist of non-SIM B constants

`MUD` - mu for normal dist of non-SIM dc constants

`MUK` - mu for normal dist of non-SIM K constants

`MUN` - mu for normal dist of non-SIM exponents

`MUP` - mu for normal dist of primary active TFs

`MUS` - mu for normal dist of supplemental active TFs

`MUBs` - mu for normal dist of SIM B constants

`MUDs` - mu for normal dist of SIM dc constants

`MUKs` - mu for normal dist of SIM K constants

`MUNs` - mu for normal dist of SIM exponents

`SIGB` - sigma for normal dist of non-SIM B constants

`SIGD` - sigma for normal dist of non-SIM dc constants

`SIGK` - sigma for normal dist of non-SIM K constants

`SIGN` - sigma for normal dist of non-SIM exponents

`SIGP` - sigma for normal dist of primary active TFs

`SIGS` - sigma for normal dist of supplemental active TFs

`SIGBs` - sigma for normal dist of SIM B constants

`SIGDs` - sigma for normal dist of SIM dc constants

`SIGKs` - sigma for normal dist of SIM K constants

`SIGNs` - sigma for normal dist of SIM exponents

`SIMon` - fraction of microarray snapshots in an experiment w/SIM(s) on

5.2 Data Generation

Introduced in Fig. 5.3 is the 100-gene network to be used in this study, with one Single Input Module (SIM) labeled, followed by the it's representation in the NEMO language.

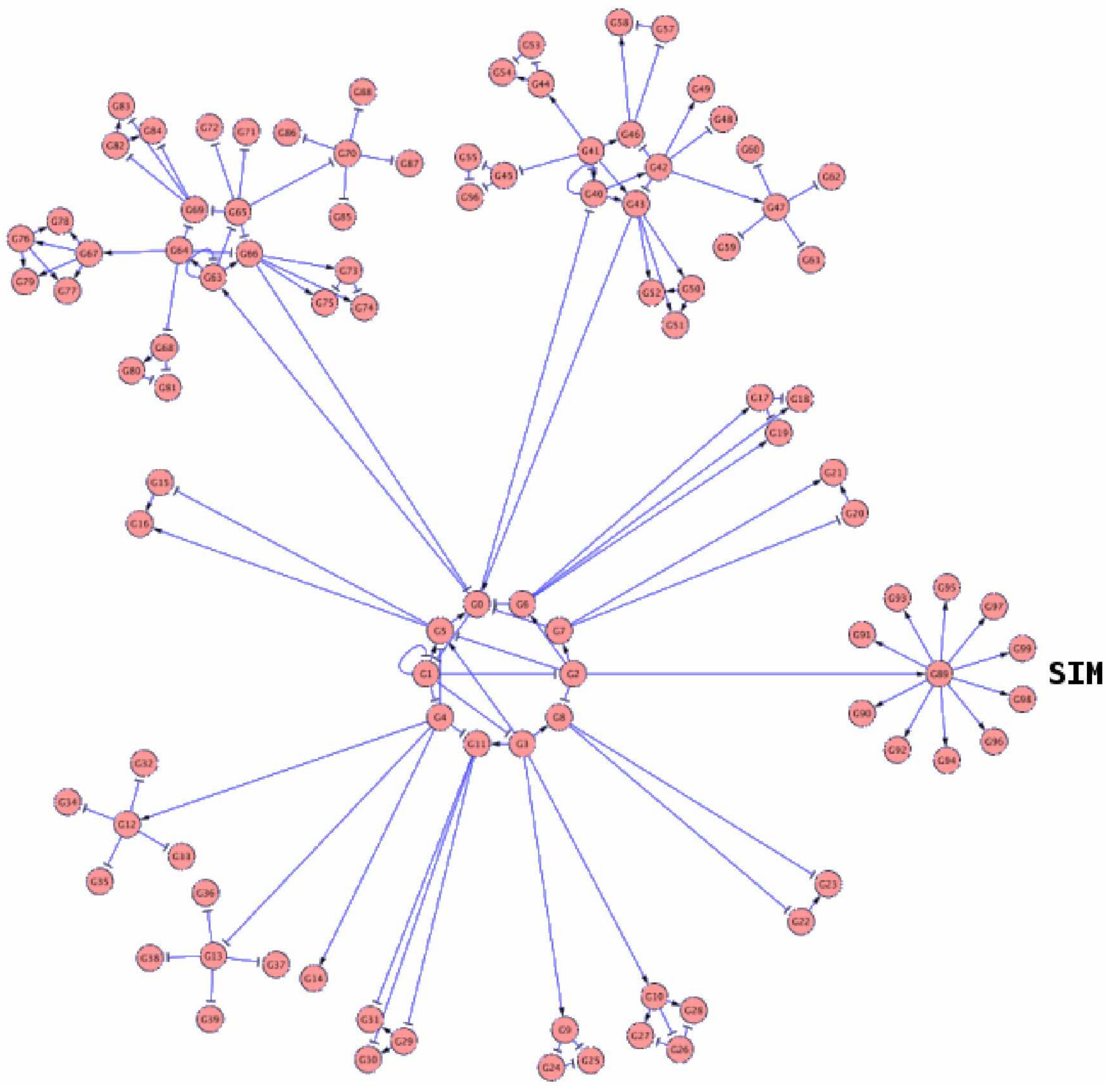


Fig. 5.3: The canonical network

```
[
GLIST(
  G1(P0-,P1-),
  G2(P1-),
  G3(P1-),
  G4(P1-),
  G40(P0-,P40+),
  G41(P40-),
  G42(P40+),
  G63(P0+,P63-),
  G64(P63+),
  G65(P63-)
)
,
DOR(
  G6(P2+),
  G7(P2+),
  G8(P2-, P3+),
  G9(P3+),
  G10(P3+),
  G11(P3+, P4-),
  G12(P4+),
  G13(P4-),
  G14(P4+),
  G5(P1+, P2-, P3+, P4-)
)
,
TMLIST(
  P5(-G15+G16+),
  P6(+G17-(G18,G19)+),
  P7(-G20+G21+),
  P8(-G22+G23-),
  P9(-G24-G25-),
  P10(-G26-(G27,G28)+),
  P11(-G29+(G30,G31)-),
  P12(-G32,G33,G34,G35),
  P13(-G36,G37,G38,G39)
)
,
```

```
DOR(
  G44(P41+),
  G45(P41-),
  G46(P41+, P42-),
  G47(P42+),
  G48(P42-),
  G49(P42+),
  G43(P40+, P41+, P42-)
)
,
TMLIST(
  P43(+G50+(G51,G52)+),
  P44(-G53-G54+),
  P45(-G55-G56-),
  P46(-G57-G58+),
  P47(-G59,G60,G61,G62)
)
,
DOR(
  G67(P64+),
  G68(P64-),
  G69(P64-, P65-),
  G70(P65-),
  G71(P65-),
  G72(P65-),
  G66(P63+, P64-, P65-)
)
,
TMLIST(
  P66(+G73-(G74,G75)+),
  P67(+G76+(G77,G78,G79)+),
  P68(+G80-G81-),
  P69(-G82+(G83,G84)-),
  P70(-G85,G86,G87,G88)
)
,
GLIST(
  G0(P5+,P6-,P7-,P43+,P66-),
  G89(P2+),
```

```
  G90(P89+),
  G91(P89+),
  G92(P89+),
  G93(P89+),
  G94(P89+),
  G95(P89+),
  G96(P89+),
  G97(P89+),
  G98(P89+),
  G99(P89+)
)
]
```

There is a reason why only one SIM module is labeled in Fig. 5.3, in spite of the fact that there are five SIM modules in the network. Close examination of the NEMO network description reveals that the switch for each of the other four modules down-regulates the members of its module, whereas for the labeled module, the switch up-regulates the members of its module. Furthermore, the members of the four modules not under consideration are initialized to zero during microarray data generation, ensuring that only he module under consideration has the possibility of being on during data generation. Later sections will examine the case where two of the five modules are up-regulated by their respective switch. This network will be referred to as the 'canonical' network, regardless of subsequent changes to its NEMO description, and references will be made to 'the canonical network configured with one module', 'the canonical network configured with two modules', etc. depending on how many modules are up-regulated by their respective switches.

When run through the COPASI biochemical simulator with all genes 'on' (every B in the Hill Functions non-zero), the typical range of times to achieve steady-state values can be observed.

A COPASI template for the network is prepared as before in section 1.3.2, but instead of editing the template by hand, a small driver written in the C language is used to pick random network parameters, which are passed to a python script that sets the parameters in the template, saving to a new file. The new file is then imported and run by COPASI, and output examined. Figure 5.4 shows COPASI output from all 100 genes.

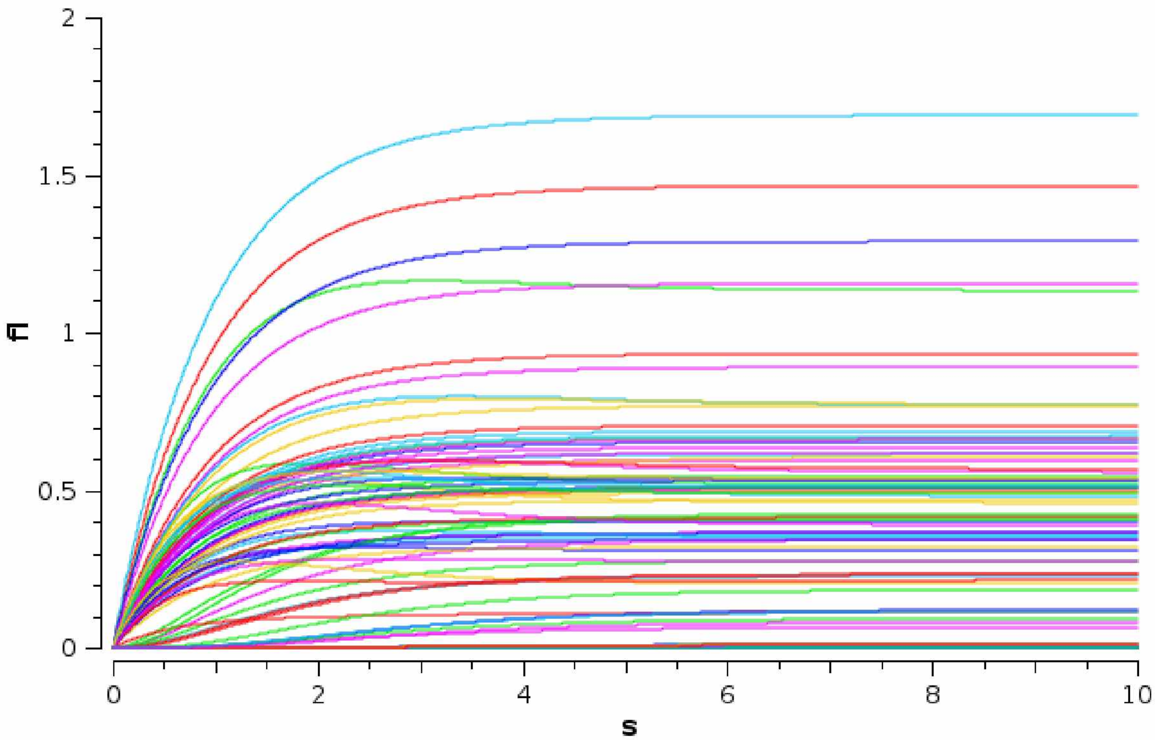


Fig. 5.4: COPASI output from 100 genes

Overall, steady-state values are substantially reached at about 5 seconds. Zooming in on just the module outputs is shown in Figure 5.5, which shows a slightly longer time steady-state time of about 8 seconds.

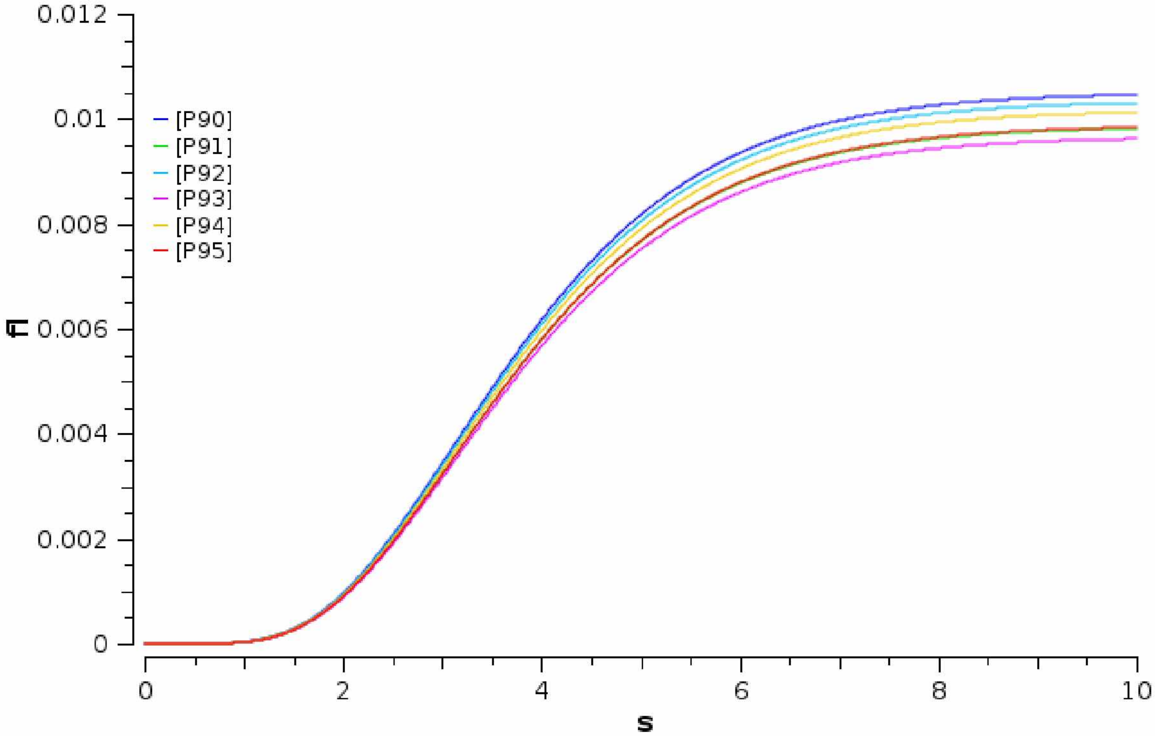


Fig. 5.5: SIM module response

The responses for SIM members in this study are assumed to be substantially similar, enforced by picking random parameter values for the SIM Hill Functions from a narrower distribution than that used for non-SIM members. Given that Pearson's correlation is invariant to separate changes in location and scale for a random variable, results will be similar for SIM genes with different asymptotes. Note that the upper bound of the module asymptotes for this particular run is not large compared to most other genes in Fig. 5.4, varying from roughly 10^{-12} to 10^{-1} femtoliters during repetitions of this experiment. Values less than 10^{-12} will be considered noise.

The network is prepared for analysis by first compiling the NEMO language representation into its SBML representation, where randomly generated Hill Functions describe the interaction between transcription factors and genes. The SBML file is subsequently saved as 'sbml100.xml':

```
$ ./nemo2sbml network.txt sbml100
$ mv sbml100_0.xml sbml100.xml
```

The compiler inserts Generalized Hill Functions with random parameters for genes whose input function is not explicitly defined, before translation of the network into SBML format. Randomly generated parameters result in most genes, usually all, being turned 'on'. Importing this representation into a biochemical simulator, and running as is, will thus generate expression data for (usually) all genes. Transcription factors, however, are generally inactive, meaning they cannot bind to DNA, and only become active by changing their shape in response to signals from outside the cell. This allosteric change in the conformation of a transcription factor allows it to bind to DNA, and thus affect the necessary genes needed to respond to the signal. In order to produce data sets that mimic allosteric signal transduction, a template of the SBML file generated above is created, with all transcription factors inactive, as outlined in section 1.3.2.

Reactivation of a subset of all factors occurs programmatically, by `generateDataSet.py`, prior to generation of a Microarray Data Set, using the following algorithm (flowchart appears in Fig. 5.6, followed by a more detailed discussion of the algorithm):

1) choose the primary active TF set (primaryTF). P1, P2, and P89 are excluded from primaryTF, as they will be turned on or off each time depending on whether the SIM is on or off.
2) for each of IDS snapshots:
3) determine if SIM is on, and set
   `primaryTFs = primaryTF + ['2','89']` if on, and
   `primaryTFs = primaryTF + ['1']` if off
4) choose supplemental active TF set (`suppTF`)
5) generate Hill Function parameters for `primaryTF`, `suppTF`, & `Gset` (genes that are not TFs, and not in SIM) from same normal distribution, and pick degradation coefficients from a uniform distribution
6) generate Hill Function parameters for SIM from its own normal distribution, and degradation coefficients from a uniform distribution
7) generate random initial conditions, keep inactive items at 0.0
8) run new cps to steady-state, grab last row, and add to data set
9) transpose the data set

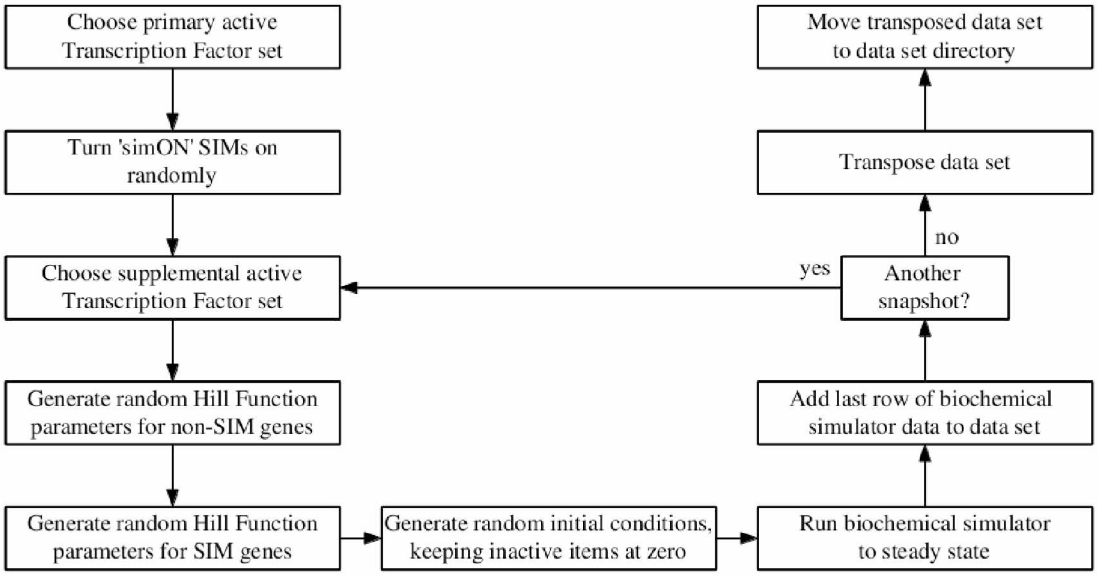


Fig. 5.6: Synthetic microarray data generation flowchart

Notes corresponding to algorithm steps above:

1) A specific state is modeled by turning on a subset of the transcription factors in the COPASI XML file. This core subset is referred to as the primary active transcription factor set. P1, P2, and P89 are turned on or off as described in step 3.
2) The number of snapshots in an experiment is 'IDS'.
3) The SIM is on when the microarray snapshot index is less than the 'fraction of snapshots on' times the current index, and off otherwise. If the SIM is on, it's switch P89 is on, and the up-regulating transcription factor for P89, viz P2, is on. The down-regulating transcription factor for P2, viz P1, is off. If the SIM is off, P2 and P89 are off, & P1 is on.
4) For each snapshot in an experiment, variation is introduced by turning on a random number of supplemental transcription factors, at least one, and no more than 20% of the number of primary transcription factors.
5) The primary and supplemental transcription factors, and all other non-SIM genes, are initialized by choosing their Hill Function parameters from the same normal distribution, where mu and sigma, along with degradation coefficients, have been chosen randomly from a uniform distribution.
6) The SIM is initialized from its own normal distribution, where the uniformly random sigma is drawn from a more narrow range than for other genes.
7) Random initial conditions are generated for all genes. Those in the SIM are randomly initialized together. One of the parameters to 'generateDataSet.py' is an integer used as an argument to 'seed()' from the python random package. Inactive genes are initialized to zero.

8) The template COPASI XML file is filled in with the new Hill Function parameters and initial conditions, and run through the simulator to steady-state. The steady-state values, representing one snapshot, are added to the experiment.

9) The snapshot data set is transposed in order to meet the pipeline input format.

A collection of Mircoarray Data Sets (experiments), suitable for each invocation of the pipeline, is generated by the bash shell script `generateDataSet.sh` that drives the python script `generateDataSet.py`, supplying for each invocation the number of data sets to generate, the number of snapshots in each set, parameters 'mu' and 'sigma' for each normal distribution, a random seed, and the fraction of microarray snapshots in a set with the SIM on. The loop used is:

1) for each of MDS states:
2) run 'generateDataSet.py' w/ MDS index, and 28 other parameters
3) append the generated data file name to 'fileList'

Notes corresponding to loop steps above:

1) The pipeline looks for groups of cellular products that have correlated expression across different organism states, where each state is represented by a series of snapshots, i.e. cellular expression levels. There are 'MDS' such states.

2) Each of the 'MDS' different states is composed of 'IDS' snapshots, and characterized by 27 other parameters.

3) Each generated state is stored in a file that is appended to 'fileList', the input file for the CODENSE algorithm.

Until now, the case of only one module in the canonical network has been discussed. The module has its own distribution from which member Hill Function parameters are picked, being distinguished from the distribution used by other genes in that its distribution sigma is smaller. When additional modules are considered, they will have their Hill Function parameters picked from the same narrower distribution, but might not be active at the same time as other modules.

5.3 Objective Functions

As stated previously, a biochemical simulator generates microarray data from a synthetic transcription network, which is then fed to the pipeline. Four objective functions compare the pipeline output to the known Single Input Modules (SIMs) contained in the synthetic transcription network, each returning either 'true' or 'false', depending upon whether or not the comparison is satisfied. The four objective functions respectively ask the following:

1) Are all known modules exactly returned, with no false-positives?
2) Are all known modules exactly returned within a set whose cardinality does not exceed `MAXMODNUM`?
3) Are at least half of all known modules exactly returned (`>= 1`) within a set whose cardinality does not exceed `MAXMODNUM`?
4) Is at least one known module approximately returned (+/- `NUMDIFF` proteins) within a set whose cardinality does not exceed `MAXMODNUM`?

where `MAXMODNUM = 10`, `NUMDIFF = 1`, and each module must have at least four members.

5.4 Anatomy of an RSA Run

An individual RSA run is composed of the following steps:

1) Initialize parameters from their respective distributions.
2) Write pipeline parameters to a file to be read at startup.
3) Pass synthetic network parameters to a python script that generates microarray data
4) Run the pipeline with the synthetic microarray data.
5) Apply objective functions to the pipeline output.
6) Update conforming and non-conforming sample (empirical) distributions (Fig. 5.1).
7) Repeat steps 1 - 6 for `NUMRUNS` iterations.
8) Write each cumulative distribution to file, compute the K-S statistic (Eqn. 5.1) between the conforming and non-conforming distributions, and output to file.

6 Regionalized Sensitivity Analysis Runs, Fixed Network Parameters

The RSA method is applied to several different pipeline configurations:

1) Using Pearson's correlation (PC) to determine microarray data set edges.

2) Using PC with Z-scores to determine microarray data set edges.

3) Using PC and Mutual Information (MI) with Z-scores to determine microarray data set edges.

Sections of the RSA driver code are configured for these different scenarios by setting different combinations of the `DO_MI` and `DO_Z` preprocessing directives.

The output from an RSA run consists of one file per parameter per objective function (Table 6.1). Each file has four columns, where the first records the conforming distribution for the objective function under consideration, the second records the non-conforming distribution for the objective function under consideration, and the third and forth columns record the cumulative distributions for the first and second columns, respectively. The third and fourth columns may be plotted as in Fig. 5.1, where the Kolmogorov-Smirnov statistic is the maximum separation between the two cumulative distributions.

Table 6.1: Example RSA output file

0	104	0.000000	0.021160
1	81	0.000197	0.037640
0	90	0.000197	0.055951
1	111	0.000393	0.078535
0	98	0.000393	0.098474
0	114	0.000590	0.121668
0	86	0.000590	0.139166
⋮	⋮	⋮	⋮

The parameter range for any given parameter is divided into one hundred equally-sized bins, resulting in columns for Table 6.1 that are one-hundred rows deep. At the end of an RSA iteration, the corresponding bin for a random parameter is incremented based upon the result of applying the objective function under consideration, i.e. the appropriate row in either the conforming or non-conforming column is incremented.

When all RSA iterations are finished, the corresponding cumulative distributions are computed, and files written to disk. An additional file for each objective function is generated that records the maximum separation between each pair of cumulative distributions for each parameter, i.e. the K-S statistic for each parameter, along with the index of the two bins where the maximum separation occurs. Table 6.2 shows a sample K-S statistic output file.

Table 6.2: Sample K-S statistic file

```
Pipeline parameters:
maxDCO   = 0.0722437942 i = 48
maxDSG   = 0.1042658489 i = 91
maxDSO   = 0.1149614546 i = 41
maxGT    = 0.0620891189 i = 14
maxPC    = 0.9909844933 i = 98
maxSOGCO = 0.0584619499 i = 26
maxSUPP  = 0.5049485114 i = 49
```

The K-S statistics and corresponding bin indices are of the most interest, as the K-S statistics tell us which parameters the pipeline is most sensitive to, and the corresponding bin indices tell us where in a parameter's distribution the K-S statistic occurs. If the cumulative conforming distribution is greater than the cumulative non-conforming distribution at this index, then the “best” parameter value for the runs lies somewhere to the left of this index, while if the cumulative conforming distribution is less than the cumulative non-conforming distribution at this index, then the “best” parameter value for the runs lies somewhere to the right of this index. In Table 6.2, the pipeline is primarily sensitive to the Pearson's correlation cutoff score, PC, and secondarily to the fraction of microarray data sets that an edge must exist in (the support, or SUPP) in order to be included in the summary graph. Since the cumulative non-conforming distribution is greater than the cumulative conforming distribution for PC, the best value of PC occurs to the right of the index, near the end of its distribution. The situation is reversed for SUPP, so that the best value for SUPP occurs occurs to the left of the index, somewhere in the first half of its distribution.

Fixed Network Parameters

The first configuration of the canonical network allows only one module the opportunity to be on, and this is reflected in the RSA driver 'sensitivitySIM.c' by setting the preprocessing variable NUMMODULES to the number of modules that might be turned on during microarray data generation:

```
#define NUMMODULES 1
```

In this chapter, we explore the behavior of the pipeline's different configurations on a data set that does not change. The RSA runs use a data collection composed of ten identical copies each of only two simple experiments, one where the single labeled module of Fig. 5.3 is turned on for 10% of the twenty snapshots in the experiment, and the other where it is turned on for 90% of the twenty snapshots in the experiment. For each experiment, all module genes have identical Hill Functions, and are initialized from a narrower distribution than other genes, under the assumption that the module will be easier to detect if its expression levels are more uniform. This assumption will be relaxed in Chapter 7, after initial investigation of pipeline behavior.

A module is considered 'on' when the transcription factor (switch) that controls the module is turned on, namely P89 for the one SIM case. To ensure that the module stays on during microarray data generation, the upstream transcription factor, P2, that up-regulates P89, is turned on, and the upstream transcription factor, P1, that down-regulates P2 is turned off. For a snapshot where the module is in the 'off' state, module genes are initialized to 0.0, along with P89 and P2, and the down-regulator transcription factor P1 for P2 is turned on.

Reiterating, ten identical copies of each simple experiment compose the data collection used for the first RSA run, for a total of twenty data sets, each with twenty snapshots. In this scenario, the RSA driver runs the exact same data collection through the pipeline for each iteration, with only pipeline parameters altered for each iteration.

The following shell command will generate ten identical '10% module on' data sets, where all variables are set to the middle of their range, the standard deviation for every normal distribution is set to zero, and the same seed (1) is passed each time to the random number generator:

```
$ for i in 0 1 2 3 4 5 6 7 8 9                                          > do
>   ./generateDataSet.py $i 20 0.4 0.055 1.0 2.5 23.0 3.0 0.55  \       >
0.055 1.0 2.5 1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.1 \       >
/home/jlong/phd/pipeline/sbml                 \               >
/home/jlong/phd/copasi/bin/CopasiSE           \               >
/home/jlong/phd/pipeline                      \               >
/home/jlong/phd/pipeline/PC_Static/OneModData \               >
/home/jlong/phd/pipeline/transpose                                  > done
```

The other set of ten identical '90% module on' data sets are generated with:

```
$ for i in 10 11 12 13 14 15 16 17 18 19                                > do
>   ./generateDataSet.py $i 20 0.4 0.055 1.0 2.5 23.0 3.0 0.55  \       >
0.055 1.0 2.5 1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.9 \             >
/home/jlong/phd/pipeline/sbml                 \                   >
/home/jlong/phd/copasi/bin/CopasiSE           \               >
/home/jlong/phd/pipeline                      \               >
/home/jlong/phd/pipeline/PC_Static/OneModData \               >
/home/jlong/phd/pipeline/transpose                                  > done
```

This particular collection of 20 snapshots is referred to as a 'static' set, since none of the Hill Function parameters will change during the RSA run on this data. For this first RSA run, we are interested in seeing if the pipeline can detect the single module labeled in Fig. 5.3 using Pearson's correlation only. There is a reason why only one SIM module is labeled in that figure, in spite of the fact that there are five SIM modules in the network. Close examination of the NEMO network description reveals that the switch for each of the other four modules down-regulates the members of its module, whereas for the labeled module, the switch up-regulates the members of its module. Furthermore, the members of the four modules not under consideration are initialized to zero during microarray data generation, ensuring that only the module under consideration has the possibility of being on during data generation. Later sections will examine the cases where more than one of the five modules is up-regulated by its switch.

The twenty snapshot files are stored in the directory 'PC_Static/OneModData', along with a 'README' file detailing how they were generated according to the instructions in section 1.3.2.

In section 5.2, steady-state for the module was about 8 seconds, so the biochemical simulator is set to run for 10 seconds. The following shell commands generate an input file telling the pipeline where to find the data:

```
$ rm fileList_STATIC
$ for i in `ls PC_Static/OneModData`
> do
>   if [ $i = README ]
>   then
>     continue
>   fi
>   echo PC_Static/OneModData/$i >> fileList_STATIC
> done
```

For a network set with fixed parameters, the RSA driver 'sensitivitySIM.c' defines preprocessing variables:

```
#define STATIC_NETWORK
#define STATIC_SEED           // use random seed of 1
#define MDS_STATIC_NETWORK 20 // number of microarray data sets
#define IDS_STATIC_NETWORK 20 // number of snapshots per
                                 microarray data set
```

With `STATIC_NETWORK` defined, the RSA driver does not generate any microarray data with random network parameters, it uses only the data with fixed parameters predefined above for each iteration. We proceed initially with number-of-iterations for an RSA run at 1000, and subsequently at 5000, 10000, and 50000, and record the K-S statistics and corresponding bin indices for each objective function in the following sections. Reiterating Section 5.3, the four objective functions are

1) Are all known modules exactly returned, with no false-positives?
2) Are all known modules exactly returned within a set with cardinality <= `MAXMODNUM`?
3) Are at least half of all known modules exactly returned (`>= 1`) within a set with cardinality <= `MAXMODNUM`?
4) Is at least one known module approximately returned (+/- `NUMDIFF` proteins) within a set with cardinality <= `MAXMODNUM`?

where `MAXMODNUM = 10`, `NUMDIFF = 1`, and each module must have at least four members.

6.1 Pearson's Correlation Only

In this section, the RSA driver 'sensitivitySIM.c' is configured to use only Pearson's Correlation for edge determination between two genes, accomplished by commenting out both the `#define DO_MI` and `#define DO_Z` statements. Results and brief commentary for each objective function are displayed separately, with an overall Discussion Summary in section 6.4.

K-S Statistics and Indices for PC, OF 1

```
              1000 runs                                  5000 runs
maxDCO   = 0.2680722892      i = 48         maxDCO   = 0.1185511895      i = 94
maxDSG   = 0.2489959839      i = 44         maxDSG   = 0.2759957231      i = 53
maxDSO   = 0.4226907631      i = 91         maxDSO   = 0.1718791767      i = 23
maxGT    = 0.2680722892      i = 48         maxGT    = 0.1780940925      i = 48
maxPC    = 0.9929718876      i = 98         maxPC    = 0.9897754611      i = 98
maxSOGCO = 0.5150602410      i = 23         maxSOGCO = 0.2489307672      i = 58
maxSUPP  = 0.8222891566      i = 17         maxSUPP  = 0.5160384924      i = 48

              10000 runs                                 50000 runs
maxDCO   = 0.1361334994      i = 48         maxDCO   = 0.0722437942      i = 48
maxDSG   = 0.0784981795      i = 73         maxDSG   = 0.1042658489      i = 91
maxDSO   = 0.0974140934      i = 56         maxDSO   = 0.1149614546      i = 41
maxGT    = 0.2687691772      i = 48         maxGT    = 0.0620891189      i = 14
maxPC    = 0.9906804289      i = 98         maxPC    = 0.9909844933      i = 98
maxSOGCO = 0.1309273283      i = 24         maxSOGCO = 0.0584619499      i = 26
maxSUPP  = 0.5181881952      i = 48         maxSUPP  = 0.5049485114      i = 49
```

Objective Function 1 (OF 1) asks the question “Are all known modules exactly returned, with no false-positives?” In this case, the pipeline ideally will find the one known module, and no others.

The two most sensitive parameters across all numbers of RSA iterations are

1) the minimum Pearson's Correlation (PC) score for an edge to be included in a 1st-order graph
2) the fraction of the microarray data sets that an edge must exist in (SUPP), in order to be included in the summary graph

The relative sensitivity of other parameters tended to decrease as the number of iterations increased. Regardless of the number of iterations, in order to return only the single known module, the PC cutoff score must be very high, due to a large number of false-positives that must be eliminated in order to detect only the known module. Trial and error settings for the range of the PC cut-off value settled in a tight range between 0.999 to 1.0 to have any chance of detecting the module, and even then, the success rate is very low, and sensitivity to the parameter very high. Examination of the output file reveals that the exact known module was found only four times during the 1000 iteration run, twelve times during the 5000 iteration run, twenty-one times during the 10000 iteration run, and eighty-six times during the 50000 iteration run, and all successes were located in the last bin.

The second observation noted is that the K-S statistic for the 'support' value (SUPP) has its maximum value at the midpoint of the SUPP range, which in this case is ½. This corresponds perfectly with the fact that our data sets have the module 'on' in ½ of the cases. Examination of the distribution of hits for SUPP shows that for values greater than ½, no module is found, reflecting the fact that, at high SUPP values, none of the module edges in the microarray data sets make it into the summary graph

For 5000 iterations and higher, the numbers for the most sensitive parameters stabilize. Knowing this number is important, because we want to have a rough idea of the minimum number of iterations that need to be used for an RSA run for the computationally expensive case where Mutual Information scores are included in the edge determination algorithm.

K-S Statistics and Indices for PC, OF 2

```
               1000 runs                                     5000 runs
maxDCO   = 0.0361438995       i = 32         maxDCO   = 0.0268981207       i = 39
maxDSG   = 0.0247103377       i = 22         maxDSG   = 0.0191678911       i = 57
maxDSO   = 0.1018476410       i = 27         maxDSO   = 0.0228839630       i = 38
maxGT    = 0.0319705086       i = 10         maxGT    = 0.0192771732       i = 25
maxPC    = 0.0560932715       i = 25         maxPC    = 0.0342299427       i = 46
maxSOGCO = 0.0501128868       i = 73         maxSOGCO = 0.0247985601       i = 81
maxSUPP  = 1.0000000000       i = 49         maxSUPP  = 1.0000000000       i = 49

               10000 runs                                    50000 runs
maxDCO   = 0.0104820374       i = 22         maxDCO   = 0.0034785164       i = 73
maxDSG   = 0.0090949301       i = 27         maxDSG   = 0.0085882135       i = 15
maxDSO   = 0.0144823728       i = 19         maxDSO   = 0.0068483920       i = 58
maxGT    = 0.0080150732       i = 26         maxGT    = 0.0029531438       i = 51
maxPC    = 0.0118198566       i = 61         maxPC    = 0.0056774532       i = 46
maxSOGCO = 0.0138525096       i = 84         maxSOGCO = 0.0050520036       i = 48
maxSUPP  = 1.0000000000       i = 49         maxSUPP  = 1.0000000000       i = 49
```

OF 2 asks the question “Are all known modules exactly returned within a set whose cardinality does not exceed `MAXMODNUM`?", where `MAXMODNUM` equals `10`. This OF admits a limited number of false-positives, among which the known module or modules must exist, and was satisfied 461 times for the 1000 runs case, 2489 times for the 5000 runs case, 4936 times for the 10000 runs case, and 24796 times for the 50000 runs case. These numbers are over two orders of magnitude greater than those for OF 1, and because the module need not be found exactly for OF 2, the pipeline sensitivity to the PC score drops dramatically.

Compared to OF 1, sensitivity to SUPP almost doubled for OF 2, from about 0.5, to 1.0. Examination of the distribution of hits shows that OF 2 was satisfied in every case where SUPP was less than or equal to ½, and not at all when above ½. Although OF 1 was also not satisfied at all when SUPP is above ½, it was also not satisfied most of the time when less than or equal to ½.

K-S Statistics and Indices for PC, OF 3

1000 runs

```
maxDCO   = 0.0361438995      i = 32
maxDSG   = 0.0247103377      i = 22
maxDSO   = 0.1018476410      i = 27
maxGT    = 0.0319705086      i = 10
maxPC    = 0.0560932715      i = 25
maxSOGCO = 0.0501128868      i = 73
maxSUPP  = 1.0000000000      i = 49
```

5000 runs

```
maxDCO   = 0.0268981207      i = 39
maxDSG   = 0.0191678911      i = 57
maxDSO   = 0.0228839630      i = 38
maxGT    = 0.0192771732      i = 25
maxPC    = 0.0342299427      i = 46
maxSOGCO = 0.0247985601      i = 81
maxSUPP  = 1.0000000000      i = 49
```

10000 runs

```
maxDCO   = 0.0104820374      i = 22
maxDSG   = 0.0090949301      i = 27
maxDSO   = 0.0144823728      i = 19
maxGT    = 0.0080150732      i = 26
maxPC    = 0.0118198566      i = 61
maxSOGCO = 0.0138525096      i = 84
maxSUPP  = 1.0000000000      i = 49
```

50000 runs

```
maxDCO   = 0.0034785164      i = 73
maxDSG   = 0.0085882135      i = 15
maxDSO   = 0.0068483920      i = 58
maxGT    = 0.0029531438      i = 51
maxPC    = 0.0056774532      i = 46
maxSOGCO = 0.0050520036      i = 48
maxSUPP  = 1.0000000000      i = 49
```

OF 3 asks the question “Are at least half of all known modules exactly returned (`>= 1`) within a set whose cardinality <= `MAXMODNUM`?", where `MAXMODNUM` equals `10`.

Because there is only one known module, OF 3 in this case is exactly the same as OF 2, and the number of times it is satisfied is identical to OF 2. For this reason, OF 3 results will be omitted in cases where only one module is turned on.

K-S Statistics and Indices for PC, OF 4

1000 runs			5000 runs		
maxDCO	= 0.0361438995	i = 32	maxDCO	= 0.0268981207	i = 39
maxDSG	= 0.0247103377	i = 22	maxDSG	= 0.0191678911	i = 57
maxDSO	= 0.1018476410	i = 27	maxDSO	= 0.0228839630	i = 38
maxGT	= 0.0319705086	i = 10	maxGT	= 0.0192771732	i = 25
maxPC	= 0.0560932715	i = 25	maxPC	= 0.0342299427	i = 46
maxSOGCO	= 0.0501128868	i = 73	maxSOGCO	= 0.0247985601	i = 81
maxSUPP	= 1.0000000000	i = 49	maxSUPP	= 1.0000000000	i = 49

10000 runs			50000 runs		
maxDCO	= 0.0104820374	i = 22	maxDCO	= 0.0034785164	i = 73
maxDSG	= 0.0090949301	i = 27	maxDSG	= 0.0085882135	i = 15
maxDSO	= 0.0144823728	i = 19	maxDSO	= 0.0068483920	i = 58
maxGT	= 0.0080150732	i = 26	maxGT	= 0.0029531438	i = 51
maxPC	= 0.0118198566	i = 61	maxPC	= 0.0056774532	i = 46
maxSOGCO	= 0.0138525096	i = 84	maxSOGCO	= 0.0050520036	i = 48
maxSUPP	= 1.0000000000	i = 49	maxSUPP	= 1.0000000000	i = 49

OF 4 ask the question "Is at least one known module approximately returned (+/- `NUMDIFF` proteins) within a set whose cardinality `<= MAXMODNUM`?" where `MAXMODNUM = 10`, `NUMDIFF = 1`, and each module must have at least four members.

OF 4 is the least restrictive OF, allowing identification of an approximate module no more than one protein off, in a set of modules possibly containing false-positives. For this particular network, OF 4 returned the exact same number of hits as OF 2 and OF 3, i.e. no approximate module was identified in those cases where the module was not exactly identified. This won't be true for every data set, so OF 4 results will continue to be shown.

6.2 Pearson's Correlation with Z-score

This section is the same as section 6.1, except that the flag to use Z-scores is uncommented:

```
#define DO_Z
```

In section 6.1, two edges are declared connected if their Pearson's correlation score exceeds a cutoff value. In this section, the high Pearson's correlation value is augmented by imposition of the additional stipulation that the correlation between two genes also be significant. As in the CLR algorithm, the distribution of correlation values in the population is first calculated, followed by application of a cutoff for the standard score, or Z-score, for each gene's correlation value. As used in the enhanced pipeline, this stipulation limits the number of false-positive edge declarations, because in addition to being significant, the correlation must also remain high. Using only the Z-score method without requiring that the correlation be high would admit the possibility of significant correlations across a low scoring distribution being declared edges.

In practice, an edge is declared if the combined Z-score exceeds a certain value, computed as

$$z_i+z_j > max(z_i) - zpc*(max(z_i) - min(z_i)) + max(z_j) - zpc*(max(z_j) - min(z_j)) \quad \text{Eqn. 6.1}$$

where z_i is the Z-score for gene *i* computed from its set of PC correlation scores, z_j is the Z-score for gene *j*, and *zpc* is the ZPC parameter of the pipeline.

The range of the PC cut-off parameter is different during a sensitivity analysis run when Z-scores are used. Due to a large number of false-positives otherwise, the PC cut-off value in the PC-only scenario of section 6.1 had a tight range between 0.999 to 1.0. In the PC with Z-score scenario, the PC cut-off score ranges between a more reasonable 0.6 to 0.9. This fact is kept in mind when comparing results between the two methods, especially in Chapter 7 when noise is added to the data.

K-S Statistics and Indices for PC, Z-score, OF 1

1000 runs

maxDCO	= 0.8261780105	i = 77
maxDSG	= 0.1887143688	i = 54
maxDSO	= 0.0833042467	i = 71
maxGT	= 0.1962769052	i = 44
maxPC	= 0.2123327516	i = 57
maxSOGCO	= 0.5162303665	i = 49
maxSUPP	= 0.6091913903	i = 40
maxZPC	= 0.0829552065	i = 12

5000 runs

maxDCO	= 0.8168689828	i = 77
maxDSG	= 0.1048027936	i = 54
maxDSO	= 0.0732354372	i = 77
maxGT	= 0.0383929346	i = 25
maxPC	= 0.1524762519	i = 64
maxSOGCO	= 0.5179991628	i = 49
maxSUPP	= 0.5232314776	i = 49
maxZPC	= 0.0487350054	i = 61

10000 runs

maxDCO	= 0.8061706370	i = 76
maxDSG	= 0.0611898999	i = 54
maxDSO	= 0.0581153653	i = 50
maxGT	= 0.0365684460	i = 70
maxPC	= 0.1495192127	i = 85
maxSOGCO	= 0.5274425438	i = 49
maxSUPP	= 0.5290166859	i = 49
maxZPC	= 0.0296707403	i = 59

50000 runs

maxDCO	= 0.8111222354	i = 77
maxDSG	= 0.0132180374	i = 51
maxDSO	= 0.0229241848	i = 62
maxGT	= 0.0137600505	i = 70
maxPC	= 0.1439848229	i = 85
maxSOGCO	= 0.5247641262	i = 49
maxSUPP	= 0.5269915317	i = 49
maxZPC	= 0.0203131544	i = 74

Even with the wider range of possible PC cut-off scores, OF 1 with the addition of Z-scores to Pearson's Correlation was more successful at finding the exact module than the simple Pearson's Correlation case, although the success rate at finding the exact module is still unacceptably low. Hits increased from 4 to 45 in the 1000 runs case, from 12 to 222 in the 5000 runs case, from 21 to 471 in the 10000 runs case, and from 86 to 2411 in the 50000 runs case. Sensitivity of the pipeline to the PC cutoff score is significantly reduced, while sensitivity is significantly increased for the density cutoff score for a coherent dense subgraph (DCO), and significantly increased for the similarity cutoff score (SOGCO) for an edge to be included in a 2nd-order graph generated from the edge support matrix.

An examination of the hit distribution for DCO shows no hits for values below index 77, for SOGCO, no hits below index 49, and for SUPP, no hits above 49.

K-S Statistics and Indices for PC, Z-score, OF 2/3

```
            1000 runs                                   5000 runs
maxDCO   = 0.0559383656      i = 57        maxDCO   = 0.0332000000      i = 60
maxDSG   = 0.0456566094      i = 92        maxDSG   = 0.0244000000      i = 53
maxDSO   = 0.0601031783      i = 51        maxDSO   = 0.0292000000      i = 19
maxGT    = 0.0261793956      i = 67        maxGT    = 0.0268000000      i = 84
maxPC    = 0.0588004217      i = 29        maxPC    = 0.0200000000      i = 60
maxSOGCO = 0.0361685326      i = 50        maxSOGCO = 0.0232000000      i = 74
maxSUPP  = 1.0000000000      i = 49        maxSUPP  = 1.0000000000      i = 49
maxZPC   = 0.0396238441      i = 48        maxZPC   = 0.0228000000      i = 34

            10000 runs                                  50000 runs
maxDCO   = 0.0211284207      i = 61        maxDCO   = 0.0070605297      i = 22
maxDSG   = 0.0111123872      i = 27        maxDSG   = 0.0085921338      i = 48
maxDSO   = 0.0277169437      i = 19        maxDSO   = 0.0073798657      i = 19
maxGT    = 0.0125686051      i = 91        maxGT    = 0.0036840768      i = 20
maxPC    = 0.0154469987      i = 23        maxPC    = 0.0103658699      i = 40
maxSOGCO = 0.0270103762      i = 68        maxSOGCO = 0.0085110370      i = 79
maxSUPP  = 1.0000000000      i = 49        maxSUPP  = 1.0000000000      i = 49
maxZPC   = 0.0134783863      i = 55        maxZPC   = 0.0062663842      i = 67
```

With the wider range of PC cut-off scores, OF 2 with the addition of Z-scores to Pearson's Correlation was only marginally more successful at being satisfied than OF 2 in the simple Pearson's Correlation only case, from 461 to 477 hits for 1000 runs, from 2489 to 2500 hits for 5000 runs, from 4936 to 4959 hits for 10000 runs, and from 24796 to 24921 hits for 50000 runs. Since OF 2 allows a limited number of false-positive coherent dense subgraphs, the sensitivities are significantly attenuated to the PC cut-off score, the density cutoff score for a coherent dense subgraph (DCO), and the similarity cutoff score (SOGCO) for an edge to be included in a 2nd-order graph.

The same total dependence on SUPP is identical to that found for the PC-only case, where OF 2 was satisfied in every case where SUPP was less than or equal to ½, and not at all when above ½.

K-S Statistics and Indices for PC, Z-score, OF 4

```
             1000 runs                                    5000 runs
maxDCO   = 0.0559383656      i = 57        maxDCO   = 0.0332000000      i = 60
maxDSG   = 0.0456566094      i = 92        maxDSG   = 0.0244000000      i = 53
maxDSO   = 0.0601031783      i = 51        maxDSO   = 0.0292000000      i = 19
maxGT    = 0.0261793956      i = 67        maxGT    = 0.0268000000      i = 84
maxPC    = 0.0588004217      i = 29        maxPC    = 0.0200000000      i = 60
maxSOGCO = 0.0361685326      i = 50        maxSOGCO = 0.0232000000      i = 74
maxSUPP  = 1.0000000000      i = 49        maxSUPP  = 1.0000000000      i = 49
maxZPC   = 0.0396238441      i = 48        maxZPC   = 0.0228000000      i = 34

             10000 runs                                   50000 runs
maxDCO   = 0.0211284207      i = 61        maxDCO   = 0.0070605297      i = 22
maxDSG   = 0.0111123872      i = 27        maxDSG   = 0.0085921338      i = 48
maxDSO   = 0.0277169437      i = 19        maxDSO   = 0.0073798657      i = 19
maxGT    = 0.0125686051      i = 91        maxGT    = 0.0036840768      i = 20
maxPC    = 0.0154469987      i = 23        maxPC    = 0.0103658699      i = 40
maxSOGCO = 0.0270103762      i = 68        maxSOGCO = 0.0085110370      i = 79
maxSUPP  = 1.0000000000      i = 49        maxSUPP  = 1.0000000000      i = 49
maxZPC   = 0.0134783863      i = 55        maxZPC   = 0.0062663842      i = 67
```

For the network with fixed parameters case, this OF returned the exact same number of hits as OF 2 and OF 3, i.e. no approximate module was identified in those cases where the module was not exactly identified among a set containing possibly false-positives. The cumulative distribution for SUPP is also unchanged.

6.3 Pearson's Correlation with Z-score and Mutual Information

Input to the pipeline is a set of graphs whose edges represent high expression correlation between two mRNA transcripts. If parameters for the routine that generates these graphs exclude real module edges often enough, the module may not be detected. In this section, we invoke the Mutual Information (MI) between two mRNA transcripts for edge determination in those cases where an edge is not declared using PC + Z-score alone. Similar to the PC + Z-score case, the MI score must exceed some cutoff, and also be significant across the distribution of MI scores for those transcripts by exceeding a Z-score cutoff.

Furthermore, the application of MI is limited by excluding cases where the sum of one mRNA's expression levels exceeds some multiple of the sum of the other mRNA's expression levels. This limitation reflects the hypothesis that many module members are expressed in quantities that are not vastly different from each other. The pipeline parameter governing which cases get excluded is SFACT, the factor by which the sum of one gene's expression levels must be less than the other for an MI calculation to proceed, and is varied during sensitivity analysis.

K-S Statistics and Indices for PC, Z-score, MI, OF 1

```
             1000 runs                                         5000 runs
maxDCO   = 0.8117770768       i = 77          maxDCO   = 0.8146515533       i = 77
maxDSG   = 0.1465267495       i = 74          maxDSG   = 0.0503031209       i = 36
maxDSO   = 0.1280714178       i = 64          maxDSO   = 0.0539462636       i = 54
maxGT    = 0.1533509303       i = 45          maxGT    = 0.0601011826       i = 39
maxPC    = 0.2007124616       i = 48          maxPC    = 0.1437867338       i = 85
maxSOGCO = 0.5457413249       i = 51          maxSOGCO = 0.5308564232       i = 49
maxSUPP  = 0.5226077813       i = 49          maxSUPP  = 0.5220403023       i = 49
maxZPC   = 0.1691667203       i = 53          maxZPC   = 0.0398184121       i = 67
maxZMI   = 0.2000472113       i = 54          maxZMI   = 0.0775875564       i = 32
maxDIM   = 0.2342110346       i = 28          maxDIM   = 0.0428389475       i = 28
maxMI    = 0.1210970192       i = 28          maxMI    = 0.0329234798       i = 56
maxNEG   = 0.1078563918       i = 67          maxNEG   = 0.0511890023       i = 74
maxPEAK  = 0.0950020387       i = 1           maxPEAK  = 0.0607558098       i = 4
maxSFACT = 0.1674070259       i = 35          maxSFACT = 0.0446605189       i = 15
```

```
             10000 runs
maxDCO   = 0.8158948779       i = 77
maxDSG   = 0.0331363527       i = 59
maxDSO   = 0.0182784564       i = 55
maxGT    = 0.0275080864       i = 71
maxPC    = 0.1441925307       i = 85
maxSOGCO = 0.5230187093       i = 49
maxSUPP  = 0.5277485810       i = 49
maxZPC   = 0.0274933799       i = 20
maxZMI   = 0.0432116067       i = 23
maxDIM   = 0.0542773872       i = 28
maxMI    = 0.0280383857       i = 29
maxNEG   = 0.0379367292       i = 74
maxPEAK  = 0.0301976468       i = 1
maxSFACT = 0.0250343656       i = 11
```

Computing an MI score for OF 1, when PC with Z-score does not infer an edge, performed slightly better at finding the exact module than the PC with Z-score case and no MI invocation, 49 vs 45 hits for 1000 runs, 236 vs 222 hits for 5000 runs, and 486 vs 471 hits for 10000 runs. The small increase in the success rate comes at high cost, due to the computation of many logarithms for MI. Cumulative distributions for DCO, SOGCO, and SUPP are the same as the PC with Z-score case.

K-S Statistics and Indices for PC, Z-score, MI, OF 2/3

1000 runs			5000 runs		
maxDCO	= 0.0396734282	i = 32	maxDCO	= 0.0122496112	i = 20
maxDSG	= 0.0751067038	i = 54	maxDSG	= 0.0202405473	i = 9
maxDSO	= 0.0381893748	i = 33	maxDSO	= 0.0212389743	i = 19
maxGT	= 0.0870991356	i = 27	maxGT	= 0.0117260771	i = 79
maxPC	= 0.0682384566	i = 84	maxPC	= 0.0230377429	i = 46
maxSOGCO	= 0.0293250557	i = 27	maxSOGCO	= 0.0213784981	i = 81
maxSUPP	= 1.0000000000	i = 49	maxSUPP	= 1.0000000000	i = 49
maxZPC	= 0.0625222508	i = 20	maxZPC	= 0.0229711811	i = 48
maxZMI	= 0.0819149489	i = 79	maxZMI	= 0.0314928516	i = 60
maxDIM	= 0.0210127565	i = 14	maxDIM	= 0.0202005462	i = 71
maxMI	= 0.0799668788	i = 50	maxMI	= 0.0250800382	i = 50
maxNEG	= 0.0665583961	i = 71	maxNEG	= 0.0183776169	i = 49
maxPEAK	= 0.0293250557	i = 1	maxPEAK	= 0.0156674636	i = 49
maxSFACT	= 0.0396814285	i = 36	maxSFACT	= 0.0468916680	i = 30

10000 runs		
maxDCO	= 0.0121331740	i = 55
maxDSG	= 0.0113979611	i = 82
maxDSO	= 0.0139371259	i = 71
maxGT	= 0.0097516120	i = 54
maxPC	= 0.0127885456	i = 46
maxSOGCO	= 0.0156029152	i = 28
maxSUPP	= 1.0000000000	i = 49
maxZPC	= 0.0105835867	i = 80
maxZMI	= 0.0126100224	i = 66
maxDIM	= 0.0077262163	i = 42
maxMI	= 0.0176197908	i = 45
maxNEG	= 0.0093155243	i = 31
maxPEAK	= 0.0066223168	i = 49
maxSFACT	= 0.0122782966	i = 27

Computing an MI score for OF 2/3, when PC with Z-score does not infer an edge, again performed only slightly better at finding the exact module, within a set of found modules that may have false-positives, than the PC with Z-score case and no MI invocation, 503 vs 477 hits for 1000 runs, 2513 vs 2500 hits for 5000 runs, and 4979 vs 4959 hits for 10000 runs. The hit distribution for SUPP shows that all hits were found when SUPP was less than or equal to ½, and none above that.

K-S Statistics and Indices for PC, Z-score, MI, OF 4

```
               1000 runs                                     5000 runs
maxDCO   = 0.0396734282      i = 32         maxDCO   = 0.0122496112      i = 20
maxDSG   = 0.0751067038      i = 54         maxDSG   = 0.0202405473      i = 9
maxDSO   = 0.0381893748      i = 33         maxDSO   = 0.0212389743      i = 19
maxGT    = 0.0870991356      i = 27         maxGT    = 0.0117260771      i = 79
maxPC    = 0.0682384566      i = 84         maxPC    = 0.0230377429      i = 46
maxSOGCO = 0.0293250557      i = 27         maxSOGCO = 0.0213784981      i = 81
maxSUPP  = 1.0000000000      i = 49         maxSUPP  = 1.0000000000      i = 49
maxZPC   = 0.0625222508      i = 20         maxZPC   = 0.0229711811      i = 48
maxZMI   = 0.0819149489      i = 79         maxZMI   = 0.0314928516      i = 60
maxDIM   = 0.0210127565      i = 14         maxDIM   = 0.0202005462      i = 71
maxMI    = 0.0799668788      i = 50         maxMI    = 0.0250800382      i = 50
maxNEG   = 0.0665583961      i = 71         maxNEG   = 0.0183776169      i = 49
maxPEAK  = 0.0293250557      i = 1          maxPEAK  = 0.0156674636      i = 49
maxSFACT = 0.0396814285      i = 36         maxSFACT = 0.0468916680      i = 30
```

```
               10000 runs
maxDCO   = 0.0121331740      i = 55
maxDSG   = 0.0113979611      i = 82
maxDSO   = 0.0139371259      i = 71
maxGT    = 0.0097516120      i = 54
maxPC    = 0.0127885456      i = 46
maxSOGCO = 0.0156029152      i = 28
maxSUPP  = 1.0000000000      i = 49
maxZPC   = 0.0105835867      i = 80
maxZMI   = 0.0126100224      i = 66
maxDIM   = 0.0077262163      i = 42
maxMI    = 0.0176197908      i = 45
maxNEG   = 0.0093155243      i = 31
maxPEAK  = 0.0066223168      i = 49
maxSFACT = 0.0122782966      i = 27
```

Once again, OF 4 returned the exact same number of hits as OF 2/3, i.e. no approximate module was identified in those cases where the module was not exactly identified among a set containing possibly false-positives. The cumulative distribution for SUPP is also unchanged.

6.4 Chapter Summary

This chapter was a preliminary look at the behavior of the pipeline for a static set of network data under sensitivity analysis where only the pipeline parameters were varied.

For OF 1, the Objective Function that determines whether or not only the exact module was found, the hit percentage during sensitivity analysis was low for all three methods, PC-only, PC with Z-score, and PC with Z-score and MI. Due to a large number of false-positives, sensitivity to the PC cut-off parameter was very high for the PC-only case, and no hits could be obtained unless the range for the PC cut-off parameter was constrained to a very tight range between 0.999 and 1.0. All three methods were also sensitive to the Support (SUPP) parameter, the fraction of microarray data sets that an edge must exist in to be included in the summary graph. The SUPP value must be set at or below the actual fraction of microarray data sets that the module edges exist in for a module to be detected.

The sensitivity to the PC cut-off parameter was significantly attenuated in the two Z-score scenarios, where the range varied between 0.6 to 0.9, the range used for all runs other than PC-only in this chapter. The trade-off is significant sensitivity to the DCO parameter, the minimum density for a dense subgraph in coherent dense subgraphs, and significant sensitivity the SOGCO parameter, the minimum similarity score for an edge to be included in a 2nd-order graph. Lower PC cut-off parameters allow more false-positive edges to enter the pipeline, which are subsequently pruned by these two parameters. An examination of the hit distributions for DCO and SOGCO show no hits below the index where their K-S statistic occurs, indicating that these cut-off scores must be sufficiently high to prune out false-positives. For parameter SUPP, no hits were recorded above where its K-S statistic occurs, again indicating that the SUPP value must be set at or below the actual fraction of microarray data sets that the module edges exist in for a module to be detected.

For OF 2, the Objective Function that allows a limited number of false-positive modules to be returned along with the exact module(s), the only significantly sensitive parameter for all three methods is SUPP. The hit percentage during sensitivity analysis increased to about 50%, with all hits occurring at SUPP values at or below ½, and no hits occurring at values above ½, consistent with the fact that the static data has the module turned on in 50% of the microarray data sets. Invoking an MI calculation, when PC with Z-score fails to infer an edge, only improved the hit percentage slightly.

OF 3 results, where at least half of all known modules are returned (>= 1) within a set containing a limited number of false-positive modules, are exactly the same for OF 2, since there is only one possible module to be returned. For this static data set, OF 4 also returned the same results as OF 2, as no approximate module was identified in those cases where the module was not exactly identified among a set containing possibly false-positives.

Chapter 7 looks at how sensitive the pipeline is to network parameters, in addition to its own parameters. We also look at how sensitive the pipeline is to noise in the data, and apply learned lessons to look at the case where the range of standard deviation for module parameters is wider ("relaxed"), and the case where two modules are 'on' in the canonical network.

7 Regionalized Sensitivity Analysis Runs, Dynamic Network

Results are presented for pipeline runs where the parameters governing the value of Hill Function constants for the transcription network are varied for each run, in addition to the pipeline parameters that were varied in the network with fixed parameters of Chapter 6. Reiterated from Section 5.1.2, they are:

`MUB` - mu for normal dist of non-SIM B constants

`MUD` - mu for normal dist of non-SIM dc constants

`MUK` - mu for normal dist of non-SIM K constants

`MUN` - mu for normal dist of non-SIM exponents

`MUP` - mu for normal dist of primary active TFs

`MUS` - mu for normal dist of supplemental active TFs

`MUBs` - mu for normal dist of SIM B constants

`MUDs` - mu for normal dist of SIM dc constants

`MUKs` - mu for normal dist of SIM K constants

`MUNs` - mu for normal dist of SIM exponents

`SIGB` - sigma for normal dist of non-SIM B constants

`SIGD` - sigma for normal dist of non-SIM dc constants

`SIGK` - sigma for normal dist of non-SIM K constants

`SIGN` - sigma for normal dist of non-SIM exponents

`SIGP` - sigma for normal dist of primary active TFs

`SIGS` - sigma for normal dist of supplemental active TFs

`SIGBs` - sigma for normal dist of SIM B constants

`SIGDs` - sigma for normal dist of SIM dc constants

`SIGKs` - sigma for normal dist of SIM K constants

`SIGNs` - sigma for normal dist of SIM exponents

`SIMon` - fraction of microarray snapshots in an experiment w/SIM(s) on

Instead of listing the K-S statistics for all parameters as in Chapter 6, an abbreviated format is used to allow easy comparison amongst results, with complete output listed in Appendix A. Interpretation of the format is described in 7.1.1 below.

7.1 One SIM

Results for one SIM active are presented for OF 1, 2, and 4 (recall that OF 3 is the same as OF 2 for one SIM), as well as the more realistic case where noise is added to the output of the biochemical simulator before entering the pipeline. The amount of noise to add is picked from a normal distribution where the mean is zero, and the standard deviation (sigma) is 10% of the value to be modified. Values of 20%, 25%, and 33% for sigma are examined in section 7.1.6.

7.1.1 Pearson's Correlation Only

The new abbreviated reporting format is as follows:

```
1) 651/5000, 1289/10000, MUB, PC, SIMon
                         MUNs, SIGKs, SIGNs
N)  15/5000,    28/10000, many are sensitive
2) 665/5000, 1305/10000, MUB, PC, SIMon
                         MUNs, SIGKs, SIGNs
N)  15/5000,    28/10000, many are sensitive
4) 729/5000, 1461/10000, MUB, PC, SIMon
                         SIGKs
N)  19/5000,    46/10000, many are sensitive
```

The number before the parenthesis indicates the Objective Function under consideration for the rest of the line, and includes the number of times the objective function was satisfied out of 5000 and 10000 runs respectively, along with the sensitive parameters whose K-S statistic is 0.2 or greater, as indicated in the 10000 run data. Parameters with a K-S statistic between 0.15 and 0.2 are listed on the next line, included for context, and referred to as 'minor' sensitivities. The “N” before a parenthesis on a line indicates the same case as the one above it, except with noise added.

For the PC case only, OF 1 was satisfied 651 times out of 5000 runs, and 1289 times out of 10,000 runs. Parameters with a K-S statistic above 0.2 are MUB, PC, and SIMon, and those with a K-S statistic between 0.15 and 0.2 are MUNs, SIGKs, and SIGNs.

For OF 1 with noise added, the hit success rate drops to about 0.3%, viz, 15 hits in 5000 runs, and 28 hits in 10,000 runs. Noise obfuscates the desired signal for all Objective Functions, resulting in many parameters being classified as sensitive (Appendix A.1.1).

This result is not surprising. Recall from Chapter 6 that for the Pearson's Correlation only algorithm, the range for the PC cut-off parameter varied in a small range between 0.999 to 1.0, due to a large number of false-positives otherwise. The Pearson's Correlation Only method depends upon finding a specific high correlation, and with the addition of noise to the data, most PC scores fall below the 0.999 cut-off, resulting in almost no edges being declared. Rerunning the Pearson's Correlation with noise case using the same PC cut-off range for Pearson's Correlation with Z-score, 0.6 to 0.9, results in a higher hit success rate (13% - 20%) for all Objective Functions in the presence of noise, as reflected in the amended reporting format:

```
1)   651/5000, 1289/10000, MUB, PC, SIMon
                           MUNs, SIGKs, SIGNs
N)   673/5000, 1310/10000, MUB, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
2)   665/5000, 1305/10000, MUB, PC, SIMon
                           MUNs, SIGKs, SIGNs
N)   693/5000, 1356/10000, MUB, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
4)   729/5000, 1461/10000, MUB, PC, SIMon
                           SIGKs
N) 1023/5000, 2025/10000, MUB, SIMon
                           PC
```

With noise added, using the new PC cut-off range, the pipeline found the exact module 673 times in 5000 runs, and 1310 times in 10000 runs. Without noise, OF 2 found the exact module in a set containing no more that 9 false-positives 665 times in 5000 runs, and 1305 times in 10000 runs. With noise, the pipeline with OF 2/3 found the exact module 693 times in 5000 runs, and 1356 times in 10000 runs. Without noise, OF 4 detected an approximate module 729 times in 5000 runs, and 1461 times in 10000 runs. With noise added, an approximate module was detected 1023 times in 5000 runs, and 2025 times in 10000 runs.

The PC-only case without noise is sensitive to the value of the PC cutoff score, even though the range is very narrow, between 0.999 to 1.0. Adding noise greatly reduces the number of false-positives, and eliminates the high sensitivity of the pipeline to the PC cutoff parameter, justifying the use of the more realistic range for the PC cutoff score of 0.6 to 0.9. Since the range is different for the 'noise' and 'no noise' cases, a direct comparison of the success rates is not meaningful. A direct comparison in subsequent sections, however, is meaningful, as the PC cutoff score ranges are identical.

An examination of the hit distributions for sensitive parameters is instructive. For PC and MUB in all cases, most hits occurred to the left of the K-S statistic index, meaning lower values are better. Lower PC cut-off values allow more edges into the pipeline, resulting in a greater chance for module edges to be detected.

The sensitivity to MUB, the mean for the normal distribution of non-module maximum expression rates (parameter B in a Hill function), is harder to explain. This could be due to a reduced number of false-positive edges between non-module and module members for values of MUB that differ sufficiently from MUBs (MUB for module members), happening more often for lower values of MUB, since its range [0.01 – 0.81) forces MUB to be often lower than MUBs, whose range is [0.3 – 0.8). It may also be due in part to the criteria of section 4.3, where it was stated that we wish to eliminate consideration of very low expression values in the data that may be considered noise. The criteria states that no edge is declared between two genes if one of the following conditions holds:

1) the sum of one gene's expression values across a data set is less than some small fraction of the other gene's expression values.
2) gene expression sums for both genes are < 10^{-12} femtoliters.
3) more than half of the expression values for one gene in a data set are < 10^{-13} femtoliters.

This hypothesis needs to be tested by counting exactly how often and at what values of MUB the criteria above are invoked.

For SIMon, no hits are recorded in bins lower than the K-S statistic index for any Objective Function, with or without noise. Recall that the pipeline parameter SUPP value must be less than or equal to the network

SIMon value in order for module edges to have a chance of being included in the summary graph. Parameter SUPP was varied from 0.3 to 0.7, while SIMon was varied from 0.1 to 1.0, so that at low values of SIMon, either SUPP was greater than SIMon, or other parameters did not cooperate to generate a hit.

Parameters in the minor sensitivity range for the non-noise case become even less sensitive with noise, all disappearing for OF 1/2/3, while for OF 4, the sensitive PC parameter drops into the minor sensitivity range, while SIGKs is eliminated from that category.

7.1.2 Pearson's Correlation with Z-score

Low hit percentages in the PC-only scenario indicate that the pipeline is not robust in the face of network variation. Z-scores are used here to eliminate edges from a vertex whose correlation score is not significant across the correlation distribution for that vertex, mitigating the number of false-positive edges.

```
1)  540/5000, 1030/10000, MUB, SIMon, ZPC
                          MUNs, SIGBs, SIGKs, SUPP
N) 1644/5000, 3235/10000, MUNs, SIMon
                          ZPC
2)  555/5000, 1064/10000, MUB, SIMon, SUPP, ZPC
                          MUNs, SIGBs, SIGKs
N) 1692/5000, 3318/10000, MUNs, SIMon
                          ZPC
4)  643/5000, 1230/10000, MUB, SIMon, ZPC
                          MUNs, SIGBs, SIGKs, SUPP
N) 2245/5000, 4350/10000, SIMon, ZPC
                          no K-S statistic between 0.15 and 0.2
```

The PC + Z-score algorithm hit rate more than doubles in the noisy case compared to the PC-only case with noise. With noise added, the number of false-positives is reduced across the distribution of PC scores, while the module PC scores are significant across that distribution.

For OF 1/2/3, the sensitive parameters in the noisy case are reduced to only MUNs, and SIMon, where higher values are better. Higher MUNs values mean higher module Hill Function exponent values, allowing the expression rates to reach equilibrium more rapidly, and higher SIMon values allow module edges to enter the pipeline more often for the values of SUPP used by the pipeline.

For OF 4, the sensitive parameters in the noisy case are reduced to only SIMon and ZPC. Higher ZPC values eliminate more false-positives, and higher SIMon values are as above.

7.1.3 Pearson's Correlation with Z-score and Mutual Information

The Pearson's Correlation with Z-score and Mutual Information (MI) case is similar to the PC with Z-score case, except that an MI score is invoked when PC + Z-score fails to infer an edge, and then only if the expression concentrations of the genes in question are not greatly different, reflecting the hypothesis that many module members are expressed in quantities that are not vastly different from each other:

```
1)   597/5000, 1207/10000, MUB, SIMon, ZPC
                           MUNs, SIGBs, SIGDs, SIGKs, SOGCO, SUPP
N) 1753/5000, 3361/10000, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
2)   621/5000, 1243/10000, MUB, SIMon, ZPC
                           MUNs, SIGBs, SIGDs, SIGKs, SOGCO, SUPP
N) 1850/5000, 3554/10000, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
4)   714/5000, 1420/10000, MUB, SIMon, ZPC
                           SIGBs, SIGKs, SOGCO, SUPP
N) 2331/5000, 4522/10000, SIMon
                           no K-S statistic between 0.15 and 0.2
```

With the addition of MI inferred module edges, the success rates for finding the module are about 5% higher than the PC + Z-score only case. The sensitivity to the pipeline parameter ZPC disappears in the presence of noise, as it did in the PC + Z-score only case, leaving MUNs and SIMon as the sensitive parameters, the same as in 7.1.2.

Also the same as in 7.1.2, many parameters in the minor sensitivity range for the noiseless case disappear in the noisy case, indicating that the algorithm is more robust in the face of variation in more realistic networks.

7.1.4 Pearson's Correlation with Z-score, Noise, and Relaxed Module

Given that the pipeline using PC with Z-score in the presence of noise is mostly insensitive to the pipeline parameter values over the ranges varied, those parameters will be set at mid-range in this section, and not vary. In addition, the module parameter standard deviations will be more relaxed, picked from a wider range that is 20% of the range for non-module parameters, in contrast to all previous cases, where the network parameter standard deviations for the SIM were picked from a distribution that was roughly 10% of the range for non-module standard deviations. This narrower distribution reflects the assumption that module members are expressed in roughly equal quantities, an assumption that is relaxed with the wider standard deviations used here.

```
1N) 1890/5000, 3745/10000, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
2N) 1898/5000, 3776/10000, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
4N) 2603/5000, 5150/10000, SIMon
                           no K-S statistic between 0.15 and 0.2
```

With the relaxed module parameters, the hit percentage increased by roughly 15% for all Objective Functions, over the noisy case when the pipeline parameters are varied with no MI computations in 7.1.2. This is probably due to the fact that the pipeline parameter that governs the fraction of microarray data sets than an edge must exist in to be included in the summary graph (SUPP) is set at 0.5, rather than varying from 0.3 to 0.7 when pipeline parameters are varied. The network parameter SIMon varies such that, on average, the SIM is on over half the time, meaning that the constant value of SUPP at 0.5 gives the pipeline a chance to detect the SIM edges more often than it would when varying SUPP values exceed SIMon values. Similar to the non-relaxed module case, the pipeline remains sensitive to MUNs, which affects the rate at which module member Hill Functions reach their equilibrium value, an assumed condition for an organism in steady-state.

7.1.5 Pearson's Correlation with Z-score, MI, Noise, and Relaxed Module

This section is the same as the previous section, except for the addition of a Mutual Information computation when PC + Z-score fails to infer an edge, and then only if the expression concentrations of the genes in question are not greatly different.

```
1N) 1862/5000, 3783/10000, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
2N) 1879/5000, 3824/10000, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
4N) 2618/5000, 5242/10000, SIMon
                           no K-S statistic between 0.15 and 0.2
```

With an MI computation included, the exact module is found only slightly more often than the PC + Z-score case alone, on the order of 1% more for the 10000 runs case, but slightly less for OF 1N and 2N of the 5000 runs case. The results in this case do not seem to justify the expense of the logarithmic computations necessary for MI, but we'll see in section 7.2.4, where two relaxed modules have the potential to be turned on, that the inclusion of an MI computation increases Objective Function hits by roughly 2%.

7.1.6 Pearson's Correlation with Z-score, MI, Various Noise Levels, and Relaxed Module

In the previous sections, runs were made with noise added to the mRNA expression levels before being admitted to stage one of the pipeline, which declares edges between correlated expression levels. The amount of noise added was randomly picked from a normal distribution with mean equal to zero, and standard deviation equal to 10% of the expression level to be modified. In this section, we re-run the scenario of section 7.1.5 (PC + Z-score, MI, and Relaxed Module) for 10000 runs with standard deviations of 20%, 25%, and 33% of the expression level to be modified, each representing more noise. As expected, the hit percentages decrease with increasing noise:

20% Standard Deviation

```
1N) 2515/10000, SIMon
                MUNs
2N) 2694/10000, SIMon
                MUNs
4N) 3818/10000, SIMon
                no K-S statistic between 0.15 and 0.2
```

With a 10% standard deviation, the hit percentage was just under 40% for OF 1/2/3, and just over 50% for Of 4, and there was a strong sensitivity to MUNs, which affects the rate at which module member Hill Functions reach equilibrium. For 20% standard deviation, the hit percentage falls to roughly 25% for OF 1/2/3, and just under 40% for OF 4. The strong sensitivity to MUNs is attenuated, making it a minor sensitivity for OF 1/2/3, and disappearing again for OF 4.

25% Standard Deviation

```
1N) 1312/10000, MUB, SIGB, SIMon
                no K-S statistic between 0.15 and 0.2
2N) 1372/10000, MUB, SIMon
                SIGB
4N) 2328/10000, SIMon
                MUB, SIGB
```

With a 25% standard deviation, hit percentages fall lower, to just under 15% for OF 1/2/3, and just under 25% for OF 4. Sensitivities to MUB and SIGB for OF 1 increase, elevating their K-S statistics to over 0.2, with almost identical results for OF 2/3. For OF 4, sensitivities to MUB and SIGB are minor.

33% Standard Deviation

```
1N) 409/10000, MUB, SIGB, SIMon
               no K-S statistic between 0.15 and 0.2
2N) 426/10000, MUB, SIGB, SIMon
               no K-S statistic between 0.15 and 0.2
4N) 830/10000, MUB, SIGB, SIMon
               IDS, MUN
```

Hit percentages fall to under 5% for OF 1/2/3, and under 10% for OF 4. Old sensitivities are mostly unchanged, with the exception for OF 4 of an increased minor sensitivity to IDS, the number of data sets in an experiment, where lower values are better, and increased sensitivity to MUN, which affects the rate at which non-module member Hill Functions reach equilibrium, where higher values result in more hits. The slight preference for lower numbers of data sets in an experiment seems counter-intuitive at first, but could simply be the case that more noisy data sets contribute to signal obfuscation.

7.2 Two SIMs

The investigation of the canonical network in preceding sections had only one Single Input Module (SIM) with the potential to be turned on. In the following sections, a second module can potentially be turned on, genes G32, G33, G34, and G35, controlled by switch G12. When the new module is turned on, G12 is turned on with its upstream up-regulator G4, while the upstream down-regulator G1 for G12 is turned off, ensuring that the module is expressed.

During synthetic microarray data generation, it is first randomly determined whether one or both of the two modules have the potential to be on for the run, with the odds of both having the potential to be on being the same as the potential for just one module to be on. After determining which modules have the potential to be on during data generation, the SIMon parameter determines whether or not the those modules are actually turned on. Both modules use normal distributions for Hill Function parameters in 7.2.1 and 7.2.2, where the maximum standard deviations are roughly 10% of those used for non-module parameters, and in 7.2.3 and 7.2.4, both modules use the relaxed parameter distributions, where the maximum standard deviations are 20% of those used for non-module parameters, and pipeline parameters are not varied.

Due to the low hit percentages of the Pearson's correlation only method, the pipeline will be configured only using PC + Z-score, or PC + Z-score with Mutual Information, with normally distributed noise added in both cases, where the mean is zero, and the standard deviation is 10% of the value to be modified.

7.2.1 Pearson's Correlation with Z-score, and Noise

```
1N) 1313/5000, 2623/10000, MUNs, SIMon, ZPC
                           SUPP
2N) 1330/5000, 2646/10000, MUNs, SIMon, ZPC
                           SUPP
3N) 1837/5000, 3587/10000, SIMon, ZPC
                           MUNs
4N) 2175/5000, 4249/10000, SIMon, ZPC
                           no K-S statistic between 0.15 and 0.2
```

For PC + Z-score with noise, the two modules case for OF 1/2 had a lower hit percentage than the one module case, about 26 % vs. 32%. For OF 3, however, the two modules case had a higher hit percentage, about 36%. Recall that for one module, OF 2 and 3 produce the same results, as OF 3 wants to know if at least half of all known modules are exactly returned within a set of restricted cardinality that may contain false-positives, and at least one module being returned. With two known modules, OF 3 is satisfied when only one of the two known modules is exactly returned within a set of restricted cardinality that may contain false-positives. OF 4 hit percentages for the one and two module cases were close, about 43% each.

In the one module case for OF 1/2, the pipeline was sensitive to MUNs, and SIMon, with lower sensitivity to ZPC, with similar results in the two modules case, except that sensitivity to ZPC increased, where higher values are better, and a minor sensitivity to SUPP manifested. For OF 3, there is reduced sensitivity to MUNs for the two module case, and for OF 4, the sensitivities are the same for both the one and two module cases.

7.2.2 Pearson's Correlation with Z-score, MI, and Noise

```
1N) 1454/5000, 2910/10000, SIMon
                           MUNs, SUPP
2N) 1483/5000, 2969/10000, SIMon
                           MUNs, SUPP
3N) 1990/5000, 3984/10000, SIMon
                           MUNs
4N) 2295/5000, 4677/10000, SIMon
                           ZPC
```

The addition of a Mutual Information calculation when PC + Z-score does not infer an edge produces better hit percentages by about 10%, and is significantly more robust, losing sensitivity to the pipeline parameter ZPC that was present in the case with no MI calculation.

Compared to the identical pipeline configuration case of 7.1.3, where only one module was allowed to be on, hit percentages for the two module case were roughly 4% lower for OF 1 and OF 2, but 4% higher for OF 3, where at least half of all known modules are returned exactly, greater than or equal to 1, within a set whose cardinality does not exceed MAXMODNUM (10). The odds of OF 3 being satisfied here are higher, where up to two modules are allowed to be on rather the one in 7.1.3. Furthermore, the sensitivity to MUNs was attenuated sufficiently to drop into the minor sensitivity category for OF 1/2/3, leaving SIMon as the only sensitive parameter.

Being sensitive to only SIMon, the fraction of microarray snapshots in an experiment with at least one SIM on, is favorable when it comes time to turn the pipeline into a tool. Only the value of SUPP will need to be adjusted downward, until it reaches the point where modules are detected. The pipeline exhibits robustness in the face of multiple modules.

7.2.3 Pearson's Correlation with Z-score, Noise, and Relaxed Modules

The pipeline is now applied to a network that represents the most realistic network so far considered, a noisy network with two relaxed modules potentially on.

```
1N) 1652/5000, 3257/10000, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
2N) 1653/5000, 3263/10000, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
3N) 2227/5000, 4437/10000, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
4N) 2626/5000, 5197/10000, SIMon
                           no K-S statistic between 0.15 and 0.2
```

Network sensitivities are comparable with section 7.2.1, the same scenario sans relaxed modules, except that minor sensitivities disappear. The detection rate is acceptable from a tool-builder's perspective, being about one-third for the two more restrictive Objective Functions, and about one-half for the two less restrictive Objective Functions. Hit percentages in section 7.2.1 are not directly comparable to those here, since pipeline parameters are fixed.

There is one important thing to note here. In section 7.2.1, the pipeline is sensitive to ZPC, a pipeline parameter, with lower values being worse. The sensitivity to ZPC there would not seem to justify running fixed pipeline parameters here, but in section 7.2.2, the sensitivity to ZPC is significantly attenuated with the addition of an MI computation. Fixed pipeline parameters are used here, with ZPC set mid-range (above the low values), so that results can be compared with the next section, where an MI computation is included.

7.2.4 Pearson's Correlation with Z-score, MI, Noise, and Relaxed Modules

This section is the same as 7.2.3, the most realistic scenario to date, except for the addition of a Mutual Information computation when PC + Z-score fails to infer an edge, and then only if the expression concentrations of the genes in question are not greatly different.

```
1N) 1877/5000, 3521/10000, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
2N) 1878/5000, 3526/10000, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
3N) 2353/5000, 4578/10000, MUNs, SIMon
                           no K-S statistic between 0.15 and 0.2
4N) 2716/5000, 5342/10000, SIMon

                           no K-S statistic between 0.15 and 0.2
```

The detection rate jumps roughly a couple of percentage points across all Objective Functions from the case in section 7.2.3, where no Mutual Information calculation was included. Section 7.2.2 showed us that sensitivity to pipeline parameters vanishes with MI included, justifying fixed pipeline parameters here.

The pipeline is only sensitive to network parameters MUNs, which affects the rate at which module gene Hill Functions reach steady-state, and SIMon, the percentage of the data sets where the modules are on. If the data represents an organism at steady-state, and the pipeline parameter SUPP parameter is set less than or equal to SIMon, then the pipeline has a good chance to detect modules.

8 Conclusions

The Network Motif (NEMO) language [Long & Roth, 2008] was developed, a translator for a qualitative transcription network description to a quantitative Systems Biology Markup Language (SBML) model. The SBML model is used as input to the COPASI biochemical simulator in order to generate synthetic gene expression data, mimicking real data that would be obtained from microarray platforms, or Next Generation Sequencing (NGS) machines.

The exponentially exact Overlapping Dense Subgraph (ODES [Long & Hartman, 2010]) algorithm was developed for finding dense subgraphs in larger graphs, even if they overlap. The main advantage of this algorithm is that it confines the brute-force search domain to the actual dense subgraphs.

An open-source version of the CODENSE pipeline was developed, derived from Zhou, et al., that features the use of the ODES algorithm, and the use of improved gene expression correlation algorithms using Z-scores and Mutual Information. An artificial NEMO transcription network with known modules was used to generate synthetic gene expression data for input to the pipeline, which identified the known modules in the data with about a 30% to 50% success rate during sensitivity analysis, depending upon desired accuracy. A manuscript describing the open-source version of the CODENSE is in preparation.

A regionalized sensitivity analysis was performed on the CODENSE algorithm, where parameters are slightly varied during each of thousands of pipeline runs, and where any discovered modules are compared to the known modules in the original network by means of a series of Objective Functions. After preliminary testing in Chapter 6 on a canonical network with fixed Hill Function and other network parameters, Chapter 7 varied the Hill Function parameters and other aspects of the network for each run during sensitivity analysis, in addition to the pipeline parameters that were varied in Chapter 6. Scenario runs progressed through configurations of the pipeline with different correlation algorithms, and combinations of either noiseless data from the COPASI simulator, or data from the simulator with different amounts of noise added.

As originally conceived, the CODENSE pipeline used only Pearson's correlation to infer graph edges between two vertices, where the vertices represent genes, and the edges connect two genes that have high expression correlation across a collection of microarray data sets. With noiseless data from the simulator, this approach was highly sensitive to the PC cut-off score used, but did better in the more realistic case where noise was added to the simulator data. The pipeline in this more realistic case lost its sensitivity to the PC cut-off score, being sensitive to how fast the module gene Hill functions reach steady-state, and sensitive to the magnitude of the equilibrium rate for non-module gene Hill Functions, MUB, with lower values resulting in more hits, where the average range of MUB for non-module members is lower than the average range for module members.

For SIMon, no hits are recorded in bins lower than the K-S statistic index for any Objective Function, with or without noise. Recall that the pipeline parameter SUPP value must be less than or equal to the network SIMon value in order for module edges to have a chance of being included in the summary graph. Parameter SUPP was varied from 0.3 to 0.7, while SIMon was varied from 0.1 to 1.0, so that at low values of SIMon, either SUPP was greater than SIMon, or other parameters did not cooperate to generate a hit.

The biggest problem with using only Pearson's correlation to infer expression correlation for genes entering the pipeline is the low success rate of the pipeline at detecting modules, due to the many false-positive expression correlations declared. The use of Z-scores and Mutual Information obtained better results, as an inferred expression correlation edge between two genes must not only have high correlation, but also be significant across the correlation distributions of both genes being connected, where significance is computed as the combined Z-score in Eqn. 6.1.

In the most realistic scenario run, where two relaxed modules had the potential to be on, the use of a Mutual Information calculation in addition to the PC + Z-scores resulted in a pipeline that was only sensitive to network parameters MUNs, which affects the rate at which module gene Hill Functions reach steady-state, and SIMon, the percentage of the data sets where the modules are on. In other words, if the data represents an organism at steady-state, and the pipeline gradually turns down the pipeline SUPP parameter until it is less than or equal to SIMon, then the pipeline has a good chance to detect modules.

9 Future Work

Several opportunities present themselves as a result of this work:

1) Package the pipeline code for use by researchers. This work is already underway.
2) Make sensitivity runs where the standard deviations for module parameters are greater than 20% of those used for non-module parameters.
3) Make sensitivity runs on the canonical network, where more than two modules have the opportunity of being turned on.
4) Make sensitivity runs on different network topologies, including those that have other motifs than the Single Input Module (SIM).
5) Experiment with different methods of initializing the overlapping dense sub-graph algorithm with edges from some subset of all the edges in the graph, i.e. from a spanning tree for the graph, or only with edges that connect vertices of high degree. This would change the algorithm from an exponentially exact algorithm to a heuristic one, perhaps with a trade-off of increased speed and the ability to handle larger dense sub-graphs, while maintaining a high probability that dense sub-graphs will be found.
6) Update the NEMO compiler to use the latest SBML library.

References

Alon, U. (2006), *An Introduction to Systems Biology: Design Principles of Biological Circuits*. 1st. Chapman & Hall/CRC. ISBN-13: 978-1584886426

Cover, T.M., Thomas, J.A. (2006), *Elements of Information Theory*. 2nd. Wiley-Interscience

Faith, J.J., Hayete, B., Thaden, J.T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J.J., Gardner, T.S. (2007), Large-Scale Mapping and Validation of *Escherichia coli* Transcriptional Regulation from a Compendium of Expression Profiles. *PLoS Biology* Vol. 5, No. 1

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., Kummer, U. (2006), COPASI - a COmplex PAthway SImulator. *Bioinformatics* 22, 3067-74

Hornberger, G.M., Cosby, B.J. (1985), Selection of Parameter Values in Environmental Models Using Sparse Data: A Case Study. *Applied Mathematics and Computation* 17:335-355

Hu, H., Yan, X., Huang, Y., Han, J., Zhou, X.J. (2005), Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics (ISMB 2005)*, Vol. 21 Suppl. 1 : i213-i221

Hucka, M., Finney, A., Bornstein, B.J., Keating, S.M., Shapiro, B.E., Matthews, J., Kovitz, B.L., Schilstra, M.J., Funahashi, A., Doyle, J.C., Kitano, H. (2004), Evolving a Lingua Franca and Associated Software Infrastructure for Computational Systems Biology: The Systems Biology Markup Language (SBML) Project. *Systems Biology* June, 1(41): 41-53

Ihaka, R., Gentleman, R. (1996): R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5, 299-314

Johnson, S.C. (1979), YACC: Yet another compiler-compiler. *Unix Programmer's Manual* Vol. 2b

Likhoshvai, V., Ratushny, A. (2007), Generalized Hill Function Method for Modeling Molecular Processes. *Journal of Bioinformatics and Computational Biology* 5 Issue 2B:521- 531

Long, J, Hartman, C. (2010), ODES: an overlapping dense sub-graph algorithm, *Bioinformatics* 26 (21): 2788-2789

Long, J., Roth, M. (2008), Synthetic mircoarray data generation with RANGE and NEMO, *Bioinformatics* 24 (1): 132-134

Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R.D., Califano, A. (2006), ARACNE: An algorithm for the reconstruction of gene regulatory networks in mammalian cellular context. *BMC Bioinformatics* 7 (Suppl 1): S7

Mendes, P., Sha, W., Ye, K. (2003): Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics* 2003, 19 Suppl 2:122-129

Ptashne, M (1992), *A Genetic Switch*. 2nd. Cell Press & Blackwell Scientific Publication

van Someren, E.P., Wessels, L.F.A., Reinders, M.J.T., Backer, E. (2001), Robust genetic network modeling by adding noisy data. *Proceedings of the 2001 IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, Baltimore, MA, June 2001.

van Someren, E.P., Wessels, L.F., Backer, E., Reinders, M.J. (2002), Genetic network modeling. *Pharmacogenomics* 3(4): 507-525.

Zephyris (2011), *http://en.wikipedia.org/wiki/File:DNA_Structure+Key+Labelled.pn_NoBB.png*, [CC-BY-SA-3.0 (www.creativecommons.org/licenses/by-sa/3.0) or GFDL (www.gnu.org/copyleft/fdl.html)], via Wikimedia Commons

Zhou, X.J., Gibson, G. (2004), Cross-species Comparison of Genome-wide Expression Patterns. *Genome Biology* 5(7), Article 232

Zhou X.J., Kao M.C., Huang H., Wong A., Nunez-Iglesias J., Primig M., Aparicio O.M., Finch C.E., Morgan T.E. & Wong W.H. (2005): *Functional annotation and network reconstruction through cross-platform integration of microarray data*. Nat Biotechnol 23: 238-43.

Appendix A: Full K-S Statistics and Indices for Dynamic Network of Chapter 7

Source code and full output from all sensitivity runs are available at

http://climate.iarc.uaf.edu/geonetwork?uuid=c0853c1a-c99e-4dbf-9a1c-4b5f6236bb5e

A.1 One SIM

A.1.1 Pearson's Correlation Only

K-S Statistics and Indices for PC, OF 1

5000 runs			10000 runs		
maxDCO	= 0.0343550559	i = 13	maxDCO	= 0.0164080104	i = 46
maxDSG	= 0.0240431704	i = 10	maxDSG	= 0.0248426345	i = 74
maxDSO	= 0.0700607764	i = 60	maxDSO	= 0.0410963943	i = 69
maxGT	= 0.0487662648	i = 38	maxGT	= 0.0235623186	i = 24
maxPC	= 0.2893268894	i = 63	maxPC	= 0.2953703703	i = 50
maxSOGCO	= 0.0710052525	i = 67	maxSOGCO	= 0.0504306950	i = 76
maxSUPP	= 0.1633498740	i = 38	maxSUPP	= 0.1358255201	i = 54
maxIDS	= 0.0929584957	i = 26	maxIDS	= 0.0901225357	i = 39
maxMDS	= 0.0639354563	i = 36	maxMDS	= 0.0170810312	i = 23
maxMUB	= 0.2430284837	i = 44	maxMUB	= 0.2190272609	i = 50
maxMUD	= 0.0200611119	i = 24	maxMUD	= 0.0184844270	i = 8
maxMUK	= 0.0709558035	i = 52	maxMUK	= 0.0181155435	i = 32
maxMUN	= 0.1293667453	i = 55	maxMUN	= 0.1233156334	i = 49
maxMUP	= 0.0293960262	i = 25	maxMUP	= 0.0567455307	i = 41
maxMUS	= 0.0416413682	i = 72	maxMUS	= 0.0215264240	i = 57
maxMUBs	= 0.0765018637	i = 35	maxMUBs	= 0.0765910503	i = 55
maxMUDs	= 0.0291176989	i = 52	maxMUDs	= 0.0327118214	i = 18
maxMUKs	= 0.0796637750	i = 37	maxMUKs	= 0.0727053949	i = 49
maxMUNs	= 0.1627201055	i = 53	maxMUNs	= 0.1566192536	i = 48
maxSIGB	= 0.1368784038	i = 60	maxSIGB	= 0.1233732547	i = 40
maxSIGD	= 0.0503341517	i = 27	maxSIGD	= 0.0194544604	i = 75
maxSIGK	= 0.0725272226	i = 49	maxSIGK	= 0.0294547463	i = 57
maxSIGN	= 0.0247668920	i = 57	maxSIGN	= 0.0241713058	i = 18
maxSIGP	= 0.0272460537	i = 85	maxSIGP	= 0.0223358836	i = 59
maxSIGS	= 0.0292826467	i = 42	maxSIGS	= 0.0371617563	i = 73
maxSIGBs	= 0.1561539828	i = 45	maxSIGBs	= 0.1480418675	i = 33
maxSIGDs	= 0.1172524432	i = 55	maxSIGDs	= 0.0828258217	i = 50
maxSIGKs	= 0.1989005365	i = 37	maxSIGKs	= 0.1847535183	i = 38
maxSIGNs	= 0.1534777315	i = 46	maxSIGNs	= 0.1648826168	i = 39
maxSIMon	= 0.5070131065	i = 43	maxSIMon	= 0.5042105881	i = 44

K-S Statistics and Indices for PC, Noise, OF 1

5000 runs			10000 runs		
maxDCO	= 0.2157138081	i = 53	maxDCO	= 0.0987335969	i = 45
maxDSG	= 0.2799063858	i = 18	maxDSG	= 0.0865423185	i = 65
maxDSO	= 0.1658976931	i = 82	maxDSO	= 0.1245773881	i = 44
maxGT	= 0.2581076563	i = 51	maxGT	= 0.1236891869	i = 22
maxPC	= 0.4541624875	i = 53	maxPC	= 0.4270672168	i = 49
maxSOGCO	= 0.2337011033	i = 15	maxSOGCO	= 0.1539310068	i = 58
maxSUPP	= 0.6057505851	i = 32	maxSUPP	= 0.6405363589	i = 21
maxIDS	= 0.3874958208	i = 53	maxIDS	= 0.2549710618	i = 36
maxMDS	= 0.2006018054	i = 16	maxMDS	= 0.1474270815	i = 33
maxMUB	= 0.4183884988	i = 31	maxMUB	= 0.3464557905	i = 61
maxMUD	= 0.1561350719	i = 49	maxMUD	= 0.1120136382	i = 35
maxMUK	= 0.4170511535	i = 37	maxMUK	= 0.2680648673	i = 51
maxMUN	= 0.2263457038	i = 63	maxMUN	= 0.2990229786	i = 32
maxMUP	= 0.3396188566	i = 73	maxMUP	= 0.2313334479	i = 22
maxMUS	= 0.1009027081	i = 89	maxMUS	= 0.1939287147	i = 76
maxMUBs	= 0.1891006352	i = 66	maxMUBs	= 0.0797088992	i = 60
maxMUDs	= 0.1903042461	i = 25	maxMUDs	= 0.1730273337	i = 56
maxMUKs	= 0.2742895353	i = 39	maxMUKs	= 0.1177296430	i = 62
maxMUNs	= 0.1862253427	i = 31	maxMUNs	= 0.1319981663	i = 55
maxSIGB	= 0.6113674356	i = 67	maxSIGB	= 0.5195117758	i = 61
maxSIGD	= 0.1519224340	i = 68	maxSIGD	= 0.2546558936	i = 38
maxSIGK	= 0.1192912070	i = 78	maxSIGK	= 0.3243223884	i = 52
maxSIGN	= 0.2510197258	i = 21	maxSIGN	= 0.1182023953	i = 93
maxSIGP	= 0.1349381478	i = 39	maxSIGP	= 0.2609449315	i = 71
maxSIGS	= 0.1720494818	i = 63	maxSIGS	= 0.1371841155	i = 36
maxSIGBs	= 0.1519224340	i = 68	maxSIGBs	= 0.1619391439	i = 79
maxSIGDs	= 0.1747910398	i = 28	maxSIGDs	= 0.1770099135	i = 63
maxSIGKs	= 0.3635573387	i = 10	maxSIGKs	= 0.2668614979	i = 26
maxSIGNs	= 0.2589769308	i = 66	maxSIGNs	= 0.1425563005	i = 74
maxSIMon	= 0.4623871615	i = 46	maxSIMon	= 0.4672081829	i = 46

Amended K-S Statistics and Indices for PC w/ new range, Noise, OF 1

5000 runs			10000 runs		
maxDCO	= 0.0220623055	i = 78	maxDCO	= 0.0360245610	i = 17
maxDSG	= 0.0316935954	i = 46	maxDSG	= 0.0298649848	i = 34
maxDSO	= 0.0526618341	i = 52	maxDSO	= 0.0329254473	i = 43
maxGT	= 0.0545051958	i = 57	maxGT	= 0.0507936647	i = 44
maxPC	= 0.0822716891	i = 83	maxPC	= 0.0996688305	i = 60
maxSOGCO	= 0.0414722031	i = 49	maxSOGCO	= 0.0423071179	i = 48
maxSUPP	= 0.0681782827	i = 79	maxSUPP	= 0.0864677307	i = 46
maxIDS	= 0.0118080912	i = 89	maxIDS	= 0.0225599311	i = 29
maxMDS	= 0.0302472021	i = 46	maxMDS	= 0.0342914116	i = 53
maxMUB	= 0.5360525207	i = 40	maxMUB	= 0.5181475593	i = 34
maxMUD	= 0.0246954830	i = 73	maxMUD	= 0.0162615624	i = 31
maxMUK	= 0.1067274802	i = 42	maxMUK	= 0.0988966874	i = 54
maxMUN	= 0.1365347892	i = 49	maxMUN	= 0.1100624566	i = 46
maxMUP	= 0.0358342911	i = 68	maxMUP	= 0.0158601182	i = 43
maxMUS	= 0.0192591458	i = 17	maxMUS	= 0.0225520252	i = 38
maxMUBs	= 0.0291596599	i = 52	maxMUBs	= 0.0278481013	i = 47
maxMUDs	= 0.0208150831	i = 64	maxMUDs	= 0.0138739799	i = 46
maxMUKs	= 0.0430930427	i = 22	maxMUKs	= 0.0543135481	i = 58
maxMUNs	= 0.2946748208	i = 40	maxMUNs	= 0.3270864994	i = 43
maxSIGB	= 0.1025902184	i = 24	maxSIGB	= 0.1177162484	i = 56
maxSIGD	= 0.0315648211	i = 25	maxSIGD	= 0.0153497483	i = 84
maxSIGK	= 0.0152187223	i = 25	maxSIGK	= 0.0229912420	i = 65
maxSIGN	= 0.0514194194	i = 42	maxSIGN	= 0.0423527965	i = 39
maxSIGP	= 0.0328137604	i = 43	maxSIGP	= 0.0233030859	i = 80
maxSIGS	= 0.0193113423	i = 78	maxSIGS	= 0.0373852546	i = 59
maxSIGBs	= 0.0206244971	i = 84	maxSIGBs	= 0.0279271603	i = 52
maxSIGDs	= 0.0122527919	i = 97	maxSIGDs	= 0.0211140295	i = 41
maxSIGKs	= 0.0189566120	i = 27	maxSIGKs	= 0.0147295742	i = 50
maxSIGNs	= 0.0185390397	i = 10	maxSIGNs	= 0.0280589253	i = 22
maxSIMon	= 0.5057912393	i = 46	maxSIMon	= 0.5152355520	i = 45

K-S Statistics and Indices for PC, OF 2/3

5000 runs			10000 runs		
maxDCO	= 0.0338909558	i = 72	maxDCO	= 0.0175875068	i = 46
maxDSG	= 0.0260044575	i = 10	maxDSG	= 0.0230819227	i = 36
maxDSO	= 0.0592276539	i = 60	maxDSO	= 0.0400454747	i = 52
maxGT	= 0.0524858860	i = 38	maxGT	= 0.0235542953	i = 21
maxPC	= 0.2913616220	i = 63	maxPC	= 0.2941048165	i = 50
maxSOGCO	= 0.0563519525	i = 67	maxSOGCO	= 0.0417468092	i = 76
maxSUPP	= 0.1736313101	i = 38	maxSUPP	= 0.1388854739	i = 54
maxIDS	= 0.0878875389	i = 26	maxIDS	= 0.0921439414	i = 39
maxMDS	= 0.0568896289	i = 49	maxMDS	= 0.0160862256	i = 23
maxMUB	= 0.2387144331	i = 45	maxMUB	= 0.2188133842	i = 42
maxMUD	= 0.0240237271	i = 24	maxMUD	= 0.0191663417	i = 18
maxMUK	= 0.0691521191	i = 52	maxMUK	= 0.0203922191	i = 31
maxMUN	= 0.1320411756	i = 55	maxMUN	= 0.1229415769	i = 49
maxMUP	= 0.0296311714	i = 81	maxMUP	= 0.0557567105	i = 41
maxMUS	= 0.0395192133	i = 72	maxMUS	= 0.0232841793	i = 57
maxMUBs	= 0.0803670075	i = 35	maxMUBs	= 0.0748873598	i = 55
maxMUDs	= 0.0349559019	i = 52	maxMUDs	= 0.0341483082	i = 18
maxMUKs	= 0.0755383268	i = 37	maxMUKs	= 0.0745039978	i = 49
maxMUNs	= 0.1627234175	i = 53	maxMUNs	= 0.1558142148	i = 48
maxSIGB	= 0.1335345284	i = 60	maxSIGB	= 0.1225022528	i = 40
maxSIGD	= 0.0513359523	i = 27	maxSIGD	= 0.0208672355	i = 75
maxSIGK	= 0.0731395964	i = 53	maxSIGK	= 0.0293769925	i = 57
maxSIGN	= 0.0259472210	i = 57	maxSIGN	= 0.0265458415	i = 18
maxSIGP	= 0.0302434286	i = 85	maxSIGP	= 0.0208844207	i = 66
maxSIGS	= 0.0305521589	i = 42	maxSIGS	= 0.0402001414	i = 73
maxSIGBs	= 0.1593065709	i = 45	maxSIGBs	= 0.1496658801	i = 39
maxSIGDs	= 0.1155050255	i = 55	maxSIGDs	= 0.0853055550	i = 50
maxSIGKs	= 0.1998785892	i = 37	maxSIGKs	= 0.1799814488	i = 38
maxSIGNs	= 0.1600159568	i = 46	maxSIGNs	= 0.1646117137	i = 39
maxSIMon	= 0.5086505190	i = 43	maxSIMon	= 0.5052368583	i = 44

K-S Statistics and Indices for PC, Noise, OF 2/3

5000 runs			10000 runs		
maxDCO	= 0.2157138081	i = 53	maxDCO	= 0.0987335969	i = 45
maxDSG	= 0.2799063858	i = 18	maxDSG	= 0.0865423185	i = 65
maxDSO	= 0.1658976931	i = 82	maxDSO	= 0.1245773881	i = 44
maxGT	= 0.2581076563	i = 51	maxGT	= 0.1236891869	i = 22
maxPC	= 0.4541624875	i = 53	maxPC	= 0.4270672168	i = 49
maxSOGCO	= 0.2337011033	i = 15	maxSOGCO	= 0.1539310068	i = 58
maxSUPP	= 0.6057505851	i = 32	maxSUPP	= 0.6405363589	i = 21
maxIDS	= 0.3874958208	i = 53	maxIDS	= 0.2549710618	i = 36
maxMDS	= 0.2006018054	i = 16	maxMDS	= 0.1474270815	i = 33
maxMUB	= 0.4183884988	i = 31	maxMUB	= 0.3464557905	i = 61
maxMUD	= 0.1561350719	i = 49	maxMUD	= 0.1120136382	i = 35
maxMUK	= 0.4170511535	i = 37	maxMUK	= 0.2680648673	i = 51
maxMUN	= 0.2263457038	i = 63	maxMUN	= 0.2990229786	i = 32
maxMUP	= 0.3396188566	i = 73	maxMUP	= 0.2313334479	i = 22
maxMUS	= 0.1009027081	i = 89	maxMUS	= 0.1939287147	i = 76
maxMUBs	= 0.1891006352	i = 66	maxMUBs	= 0.0797088992	i = 60
maxMUDs	= 0.1903042461	i = 25	maxMUDs	= 0.1730273337	i = 56
maxMUKs	= 0.2742895353	i = 39	maxMUKs	= 0.1177296430	i = 62
maxMUNs	= 0.1862253427	i = 31	maxMUNs	= 0.1319981663	i = 55
maxSIGB	= 0.6113674356	i = 67	maxSIGB	= 0.5195117758	i = 61
maxSIGD	= 0.1519224340	i = 68	maxSIGD	= 0.2546558936	i = 38
maxSIGK	= 0.1192912070	i = 78	maxSIGK	= 0.3243223884	i = 52
maxSIGN	= 0.2510197258	i = 21	maxSIGN	= 0.1182023953	i = 93
maxSIGP	= 0.1349381478	i = 39	maxSIGP	= 0.2609449315	i = 71
maxSIGS	= 0.1720494818	i = 63	maxSIGS	= 0.1371841155	i = 36
maxSIGBs	= 0.1519224340	i = 68	maxSIGBs	= 0.1619391439	i = 79
maxSIGDs	= 0.1747910398	i = 28	maxSIGDs	= 0.1770099135	i = 63
maxSIGKs	= 0.3635573387	i = 10	maxSIGKs	= 0.2668614979	i = 26
maxSIGNs	= 0.2589769308	i = 66	maxSIGNs	= 0.1425563005	i = 74
maxSIMon	= 0.4623871615	i = 46	maxSIMon	= 0.4672081829	i = 46

Amended K-S Statistics and Indices for PC w/ new range, Noise, OF 2/3

5000 runs			10000 runs		
maxDCO	= 0.0217864070	i = 78	maxDCO	= 0.0385542037	i = 31
maxDSG	= 0.0337897533	i = 46	maxDSG	= 0.0306321912	i = 34
maxDSO	= 0.0489337302	i = 52	maxDSO	= 0.0296746153	i = 45
maxGT	= 0.0572051069	i = 57	maxGT	= 0.0513108484	i = 34
maxPC	= 0.0782380172	i = 83	maxPC	= 0.0981047778	i = 60
maxSOGCO	= 0.0470960559	i = 49	maxSOGCO	= 0.0325456367	i = 42
maxSUPP	= 0.0630337338	i = 79	maxSUPP	= 0.0798634004	i = 74
maxIDS	= 0.0144698838	i = 63	maxIDS	= 0.0273547290	i = 29
maxMDS	= 0.0357691479	i = 46	maxMDS	= 0.0337496024	i = 73
maxMUB	= 0.5394228865	i = 40	maxMUB	= 0.5122744441	i = 34
maxMUD	= 0.0241035182	i = 75	maxMUD	= 0.0140240848	i = 95
maxMUK	= 0.1083232739	i = 42	maxMUK	= 0.0971318452	i = 54
maxMUN	= 0.1212359088	i = 49	maxMUN	= 0.0957504242	i = 41
maxMUP	= 0.0362867790	i = 60	maxMUP	= 0.0164811577	i = 43
maxMUS	= 0.0203872953	i = 59	maxMUS	= 0.0209318722	i = 38
maxMUBs	= 0.0308283673	i = 52	maxMUBs	= 0.0211710955	i = 47
maxMUDs	= 0.0186665487	i = 64	maxMUDs	= 0.0109926711	i = 94
maxMUKs	= 0.0445899842	i = 22	maxMUKs	= 0.0569327677	i = 54
maxMUNs	= 0.3015460921	i = 45	maxMUNs	= 0.3323491391	i = 43
maxSIGB	= 0.1118183728	i = 24	maxSIGB	= 0.1239057494	i = 56
maxSIGD	= 0.0303328485	i = 30	maxSIGD	= 0.0129358062	i = 84
maxSIGK	= 0.0142119058	i = 11	maxSIGK	= 0.0211025023	i = 65
maxSIGN	= 0.0412486670	i = 42	maxSIGN	= 0.0354794500	i = 39
maxSIGP	= 0.0305797703	i = 38	maxSIGP	= 0.0228070966	i = 80
maxSIGS	= 0.0189335727	i = 63	maxSIGS	= 0.0367779448	i = 59
maxSIGBs	= 0.0215964414	i = 84	maxSIGBs	= 0.0265322921	i = 52
maxSIGDs	= 0.0142980101	i = 29	maxSIGDs	= 0.0250870555	i = 51
maxSIGKs	= 0.0192387908	i = 27	maxSIGKs	= 0.0148079593	i = 50
maxSIGNs	= 0.0175448471	i = 10	maxSIGNs	= 0.0319412650	i = 22
maxSIMon	= 0.5088866709	i = 46	maxSIMon	= 0.5183369302	i = 45

K-S Statistics and Indices for PC, OF 4

5000 runs			10000 runs		
maxDCO	= 0.0258928769	i = 12	maxDCO	= 0.0320467855	i = 76
maxDSG	= 0.0282342490	i = 10	maxDSG	= 0.0222747359	i = 57
maxDSO	= 0.0565025426	i = 60	maxDSO	= 0.0347625129	i = 52
maxGT	= 0.0381762478	i = 34	maxGT	= 0.0303139463	i = 24
maxPC	= 0.2819339540	i = 63	maxPC	= 0.2848544733	i = 49
maxSOGCO	= 0.0482952146	i = 67	maxSOGCO	= 0.0404497495	i = 71
maxSUPP	= 0.1579067556	i = 48	maxSUPP	= 0.1379147045	i = 40
maxIDS	= 0.0874793123	i = 33	maxIDS	= 0.0880656366	i = 46
maxMDS	= 0.0466989705	i = 49	maxMDS	= 0.0178062101	i = 39
maxMUB	= 0.2236077749	i = 45	maxMUB	= 0.2162140628	i = 42
maxMUD	= 0.0171321629	i = 23	maxMUD	= 0.0139832707	i = 8
maxMUK	= 0.0633712096	i = 52	maxMUK	= 0.0243750160	i = 32
maxMUN	= 0.1399289366	i = 55	maxMUN	= 0.1209254571	i = 49
maxMUP	= 0.0246194146	i = 25	maxMUP	= 0.0498044203	i = 41
maxMUS	= 0.0328257791	i = 72	maxMUS	= 0.0157141060	i = 57
maxMUBs	= 0.0694921792	i = 35	maxMUBs	= 0.0717206129	i = 55
maxMUDs	= 0.0324394046	i = 52	maxMUDs	= 0.0285917679	i = 18
maxMUKs	= 0.0729634479	i = 37	maxMUKs	= 0.0678420444	i = 50
maxMUNs	= 0.1646867781	i = 42	maxMUNs	= 0.1420710179	i = 48
maxSIGB	= 0.1345309339	i = 60	maxSIGB	= 0.1283482582	i = 40
maxSIGD	= 0.0413591006	i = 27	maxSIGD	= 0.0232901679	i = 75
maxSIGK	= 0.0720978790	i = 49	maxSIGK	= 0.0405945936	i = 55
maxSIGN	= 0.0242423542	i = 57	maxSIGN	= 0.0201870405	i = 18
maxSIGP	= 0.0262339657	i = 85	maxSIGP	= 0.0145401231	i = 66
maxSIGS	= 0.0211960011	i = 42	maxSIGS	= 0.0317215876	i = 75
maxSIGBs	= 0.1538984808	i = 49	maxSIGBs	= 0.1423817073	i = 44
maxSIGDs	= 0.1014058189	i = 50	maxSIGDs	= 0.0743852801	i = 50
maxSIGKs	= 0.1905000034	i = 37	maxSIGKs	= 0.1806024442	i = 45
maxSIGNs	= 0.1619863314	i = 46	maxSIGNs	= 0.1481601628	i = 39
maxSIMon	= 0.5162725357	i = 43	maxSIMon	= 0.5145278991	i = 44

K-S Statistics and Indices for PC, Noise, OF 4

	5000 runs			10000 runs	
maxDCO	= 0.1897315060	i = 49	maxDCO	= 0.1151601716	i = 86
maxDSG	= 0.2343431355	i = 18	maxDSG	= 0.0673969826	i = 24
maxDSO	= 0.1170130707	i = 16	maxDSO	= 0.1338111836	i = 52
maxGT	= 0.2289753696	i = 37	maxGT	= 0.1145530309	i = 22
maxPC	= 0.4545272034	i = 53	maxPC	= 0.4041416603	i = 52
maxSOGCO	= 0.2429759402	i = 16	maxSOGCO	= 0.1245992435	i = 71
maxSUPP	= 0.6286626021	i = 16	maxSUPP	= 0.5403726708	i = 28
maxIDS	= 0.4018956244	i = 53	maxIDS	= 0.3102619878	i = 29
maxMDS	= 0.1750335485	i = 76	maxMDS	= 0.0947445204	i = 79
maxMUB	= 0.3165819588	i = 31	maxMUB	= 0.2970184588	i = 39
maxMUD	= 0.1546930969	i = 31	maxMUD	= 0.0604083130	i = 31
maxMUK	= 0.4068195987	i = 37	maxMUK	= 0.2738903303	i = 46
maxMUN	= 0.2106742463	i = 41	maxMUN	= 0.2337142158	i = 70
maxMUP	= 0.3187586513	i = 73	maxMUP	= 0.1584331403	i = 22
maxMUS	= 0.1695706844	i = 3	maxMUS	= 0.1297621232	i = 73
maxMUBs	= 0.1822081806	i = 66	maxMUBs	= 0.0860785701	i = 64
maxMUDs	= 0.1517133528	i = 25	maxMUDs	= 0.1093814154	i = 56
maxMUKs	= 0.2724775198	i = 46	maxMUKs	= 0.1070227394	i = 62
maxMUNs	= 0.2145521402	i = 31	maxMUNs	= 0.1362310105	i = 24
maxSIGB	= 0.5669227274	i = 61	maxSIGB	= 0.4970516550	i = 61
maxSIGD	= 0.1180168852	i = 82	maxSIGD	= 0.1907033222	i = 33
maxSIGK	= 0.1017762233	i = 78	maxSIGK	= 0.2347799879	i = 71
maxSIGN	= 0.2582656199	i = 21	maxSIGN	= 0.1355758227	i = 93
maxSIGP	= 0.1914010080	i = 39	maxSIGP	= 0.1381791021	i = 71
maxSIGS	= 0.1651433341	i = 63	maxSIGS	= 0.0932637961	i = 46
maxSIGBs	= 0.1062564059	i = 68	maxSIGBs	= 0.1017157184	i = 19
maxSIGDs	= 0.1643931149	i = 77	maxSIGDs	= 0.1851298582	i = 63
maxSIGKs	= 0.2652289225	i = 10	maxSIGKs	= 0.1222667750	i = 26
maxSIGNs	= 0.1566161942	i = 21	maxSIGNs	= 0.1387250919	i = 68
maxSIMon	= 0.4627584822	i = 46	maxSIMon	= 0.4582077557	i = 45

Amended K-S Statistics and Indices for PC w/ new range, Noise, OF 4

5000 runs			10000 runs		
maxDCO	= 0.0204833869	i = 26	maxDCO	= 0.0263230001	i = 31
maxDSG	= 0.0268483316	i = 66	maxDSG	= 0.0360664112	i = 34
maxDSO	= 0.0276570819	i = 52	maxDSO	= 0.0360772476	i = 46
maxGT	= 0.0310248929	i = 57	maxGT	= 0.0269406711	i = 34
maxPC	= 0.1513542604	i = 64	maxPC	= 0.1541267077	i = 60
maxSOGCO	= 0.0325799533	i = 36	maxSOGCO	= 0.0265939084	i = 44
maxSUPP	= 0.0926276854	i = 55	maxSUPP	= 0.1116002941	i = 48
maxIDS	= 0.0364918867	i = 63	maxIDS	= 0.0228971709	i = 49
maxMDS	= 0.0263408803	i = 86	maxMDS	= 0.0347211579	i = 46
maxMUB	= 0.4811281671	i = 41	maxMUB	= 0.4593892953	i = 34
maxMUD	= 0.0139400415	i = 73	maxMUD	= 0.0159820426	i = 71
maxMUK	= 0.0945306275	i = 42	maxMUK	= 0.0900282519	i = 56
maxMUN	= 0.1257616724	i = 49	maxMUN	= 0.1102256279	i = 65
maxMUP	= 0.0291506932	i = 56	maxMUP	= 0.0218506908	i = 44
maxMUS	= 0.0153949992	i = 14	maxMUS	= 0.0274360463	i = 38
maxMUBs	= 0.0227847574	i = 51	maxMUBs	= 0.0174062464	i = 33
maxMUDs	= 0.0167221031	i = 51	maxMUDs	= 0.0112140563	i = 65
maxMUKs	= 0.0290882529	i = 15	maxMUKs	= 0.0270118813	i = 54
maxMUNs	= 0.1017538782	i = 63	maxMUNs	= 0.0960780216	i = 52
maxSIGB	= 0.0744040422	i = 27	maxSIGB	= 0.0920391656	i = 49
maxSIGD	= 0.0160560734	i = 32	maxSIGD	= 0.0207113278	i = 72
maxSIGK	= 0.0330715467	i = 78	maxSIGK	= 0.0234854290	i = 50
maxSIGN	= 0.0587959531	i = 67	maxSIGN	= 0.0330539108	i = 43
maxSIGP	= 0.0302547960	i = 7	maxSIGP	= 0.0162204420	i = 9
maxSIGS	= 0.0240484299	i = 81	maxSIGS	= 0.0243399512	i = 62
maxSIGBs	= 0.0127586316	i = 75	maxSIGBs	= 0.0224265645	i = 82
maxSIGDs	= 0.0224794938	i = 63	maxSIGDs	= 0.0232764426	i = 41
maxSIGKs	= 0.0189957223	i = 74	maxSIGKs	= 0.0182685088	i = 52
maxSIGNs	= 0.0214923426	i = 52	maxSIGNs	= 0.0180811951	i = 24
maxSIMon	= 0.5451800064	i = 44	maxSIMon	= 0.5599256937	i = 46

A.1.2 Pearson's Correlation with Z-score

K-S Statistics and Indices for PC, Z-score, OF 1

5000 runs			10000 runs		
maxDCO	= 0.0255854509	i = 79	maxDCO	= 0.0315950688	i = 64
maxDSG	= 0.0321873443	i = 41	maxDSG	= 0.0270610774	i = 54
maxDSO	= 0.0362564358	i = 20	maxDSO	= 0.0230206405	i = 35
maxGT	= 0.0267646570	i = 54	maxGT	= 0.0197659945	i = 39
maxPC	= 0.0351602724	i = 68	maxPC	= 0.0274290786	i = 42
maxSOGCO	= 0.1537286165	i = 53	maxSOGCO	= 0.1375761708	i = 61
maxSUPP	= 0.1651469856	i = 46	maxSUPP	= 0.1967312834	i = 45
maxZPC	= 0.2836239827	i = 34	maxZPC	= 0.2807221483	i = 45
maxIDS	= 0.0545839562	i = 56	maxIDS	= 0.0573616478	i = 19
maxMDS	= 0.0570171068	i = 53	maxMDS	= 0.0421794331	i = 26
maxMUB	= 0.2595582129	i = 38	maxMUB	= 0.2325486249	i = 39
maxMUD	= 0.0305264906	i = 44	maxMUD	= 0.0339935708	i = 71
maxMUK	= 0.0466782926	i = 45	maxMUK	= 0.0473963914	i = 28
maxMUN	= 0.1293638930	i = 46	maxMUN	= 0.1342165362	i = 70
maxMUP	= 0.0404999170	i = 64	maxMUP	= 0.0249732117	i = 29
maxMUS	= 0.0356585285	i = 31	maxMUS	= 0.0303936531	i = 60
maxMUBs	= 0.0472429829	i = 44	maxMUBs	= 0.0481789352	i = 64
maxMUDs	= 0.0588357416	i = 37	maxMUDs	= 0.0170319620	i = 74
maxMUKs	= 0.0743148979	i = 77	maxMUKs	= 0.0851760453	i = 49
maxMUNs	= 0.1562115928	i = 44	maxMUNs	= 0.1656633222	i = 54
maxSIGB	= 0.0869789072	i = 54	maxSIGB	= 0.0840049355	i = 61
maxSIGD	= 0.0368128218	i = 34	maxSIGD	= 0.0170633503	i = 67
maxSIGK	= 0.0620993190	i = 15	maxSIGK	= 0.0471810025	i = 82
maxSIGN	= 0.0516940708	i = 55	maxSIGN	= 0.0152763797	i = 63
maxSIGP	= 0.0339229364	i = 67	maxSIGP	= 0.0164193482	i = 33
maxSIGS	= 0.0378425511	i = 72	maxSIGS	= 0.0337619465	i = 48
maxSIGBs	= 0.1221890051	i = 52	maxSIGBs	= 0.1659003583	i = 45
maxSIGDs	= 0.1461052981	i = 58	maxSIGDs	= 0.1146312953	i = 46
maxSIGKs	= 0.1933648896	i = 41	maxSIGKs	= 0.1759511208	i = 30
maxSIGNs	= 0.1349526657	i = 40	maxSIGNs	= 0.1375999827	i = 32
maxSIMon	= 0.5043099153	i = 44	maxSIMon	= 0.4985453129	i = 45

K-S Statistics and Indices for PC, Z-score, Noise, OF 1

5000 runs			10000 runs		
maxDCO	= 0.0248717480	i = 20	maxDCO	= 0.0128964086	i = 13
maxDSG	= 0.0348832320	i = 48	maxDSG	= 0.0109934875	i = 39
maxDSO	= 0.0347106827	i = 49	maxDSO	= 0.0164523053	i = 47
maxGT	= 0.0211206134	i = 79	maxGT	= 0.0102564454	i = 31
maxPC	= 0.0237559776	i = 69	maxPC	= 0.0283939405	i = 42
maxSOGCO	= 0.0459771365	i = 59	maxSOGCO	= 0.0408471186	i = 53
maxSUPP	= 0.0191609464	i = 21	maxSUPP	= 0.0372235949	i = 41
maxZPC	= 0.1624950338	i = 18	maxZPC	= 0.1652913955	i = 19
maxIDS	= 0.0694866151	i = 43	maxIDS	= 0.0381863647	i = 23
maxMDS	= 0.0411675062	i = 23	maxMDS	= 0.0271394154	i = 43
maxMUB	= 0.0633872151	i = 58	maxMUB	= 0.0478156618	i = 67
maxMUD	= 0.0268422899	i = 73	maxMUD	= 0.0168605800	i = 45
maxMUK	= 0.0274933373	i = 69	maxMUK	= 0.0151936677	i = 14
maxMUN	= 0.0176399027	i = 23	maxMUN	= 0.0113846270	i = 28
maxMUP	= 0.0151430129	i = 26	maxMUP	= 0.0140913032	i = 64
maxMUS	= 0.0172150544	i = 29	maxMUS	= 0.0142462054	i = 9
maxMUBs	= 0.0239524518	i = 53	maxMUBs	= 0.0115852231	i = 39
maxMUDs	= 0.0256184950	i = 62	maxMUDs	= 0.0289523196	i = 46
maxMUKs	= 0.0227895566	i = 51	maxMUKs	= 0.0188455216	i = 57
maxMUNs	= 0.2406758132	i = 32	maxMUNs	= 0.2391818056	i = 28
maxSIGB	= 0.0103210577	i = 85	maxSIGB	= 0.0228409933	i = 43
maxSIGD	= 0.0255670202	i = 36	maxSIGD	= 0.0118900012	i = 46
maxSIGK	= 0.0198721685	i = 31	maxSIGK	= 0.0425910250	i = 37
maxSIGN	= 0.0202332170	i = 72	maxSIGN	= 0.0128559695	i = 34
maxSIGP	= 0.0293877545	i = 43	maxSIGP	= 0.0261588250	i = 54
maxSIGS	= 0.0150052635	i = 51	maxSIGS	= 0.0173033536	i = 34
maxSIGBs	= 0.0174064536	i = 64	maxSIGBs	= 0.0151482023	i = 60
maxSIGDs	= 0.0226315072	i = 48	maxSIGDs	= 0.0117446947	i = 38
maxSIGKs	= 0.0280994348	i = 53	maxSIGKs	= 0.0236669100	i = 40
maxSIGNs	= 0.0159426846	i = 47	maxSIGNs	= 0.0210354002	i = 20
maxSIMon	= 0.6569729242	i = 45	maxSIMon	= 0.6579578725	i = 44

K-S Statistics and Indices for PC, Z-score, OF 2/3

5000 runs			10000 runs		
maxDCO	= 0.0274425156	i = 76	maxDCO	= 0.0389999731	i = 64
maxDSG	= 0.0344916345	i = 41	maxDSG	= 0.0239985595	i = 54
maxDSO	= 0.0310866547	i = 20	maxDSO	= 0.0187507152	i = 24
maxGT	= 0.0276431662	i = 54	maxGT	= 0.0184116289	i = 39
maxPC	= 0.0379249891	i = 68	maxPC	= 0.0263040098	i = 42
maxSOGCO	= 0.1377517000	i = 53	maxSOGCO	= 0.1283328061	i = 64
maxSUPP	= 0.1714468124	i = 46	maxSUPP	= 0.2033104247	i = 45
maxZPC	= 0.2858196779	i = 34	maxZPC	= 0.2810141962	i = 45
maxIDS	= 0.0596661904	i = 56	maxIDS	= 0.0527704108	i = 19
maxMDS	= 0.0555417059	i = 66	maxMDS	= 0.0363444982	i = 36
maxMUB	= 0.2616726963	i = 38	maxMUB	= 0.2254074084	i = 39
maxMUD	= 0.0322824512	i = 86	maxMUD	= 0.0313229919	i = 71
maxMUK	= 0.0411495860	i = 45	maxMUK	= 0.0481199642	i = 28
maxMUN	= 0.1285055584	i = 46	maxMUN	= 0.1304548300	i = 70
maxMUP	= 0.0322479960	i = 64	maxMUP	= 0.0244453457	i = 29
maxMUS	= 0.0350469705	i = 31	maxMUS	= 0.0293574693	i = 60
maxMUBs	= 0.0416035833	i = 44	maxMUBs	= 0.0435233675	i = 61
maxMUDs	= 0.0601039735	i = 37	maxMUDs	= 0.0192227435	i = 74
maxMUKs	= 0.0698365407	i = 77	maxMUKs	= 0.0852436036	i = 52
maxMUNs	= 0.1539800768	i = 44	maxMUNs	= 0.1637694280	i = 54
maxSIGB	= 0.0851893513	i = 54	maxSIGB	= 0.0847656855	i = 61
maxSIGD	= 0.0474650128	i = 34	maxSIGD	= 0.0112874509	i = 67
maxSIGK	= 0.0594939146	i = 15	maxSIGK	= 0.0450160204	i = 82
maxSIGN	= 0.0515307208	i = 55	maxSIGN	= 0.0168314699	i = 63
maxSIGP	= 0.0327425288	i = 67	maxSIGP	= 0.0139698508	i = 83
maxSIGS	= 0.0388147428	i = 72	maxSIGS	= 0.0348165064	i = 25
maxSIGBs	= 0.1172529110	i = 52	maxSIGBs	= 0.1660067245	i = 45
maxSIGDs	= 0.1426098765	i = 63	maxSIGDs	= 0.1121164034	i = 46
maxSIGKs	= 0.1945804072	i = 41	maxSIGKs	= 0.1785402966	i = 30
maxSIGNs	= 0.1317382624	i = 39	maxSIGNs	= 0.1387729619	i = 32
maxSIMon	= 0.5061806463	i = 44	maxSIMon	= 0.5010324042	i = 45

K-S Statistics and Indices for PC, Z-score, Noise, OF 2/3

5000 runs			10000 runs		
maxDCO	= 0.0253043699	i = 20	maxDCO	= 0.0139154628	i = 13
maxDSG	= 0.0305120333	i = 48	maxDSG	= 0.0140766653	i = 39
maxDSO	= 0.0262355605	i = 49	maxDSO	= 0.0180206682	i = 47
maxGT	= 0.0212687346	i = 56	maxGT	= 0.0109083646	i = 31
maxPC	= 0.0219619463	i = 69	maxPC	= 0.0254715240	i = 42
maxSOGCO	= 0.0347899354	i = 59	maxSOGCO	= 0.0337778264	i = 53
maxSUPP	= 0.0283452108	i = 27	maxSUPP	= 0.0319357702	i = 41
maxZPC	= 0.1594929978	i = 18	maxZPC	= 0.1632755512	i = 19
maxIDS	= 0.0659565892	i = 43	maxIDS	= 0.0383301950	i = 29
maxMDS	= 0.0410345577	i = 23	maxMDS	= 0.0231171741	i = 79
maxMUB	= 0.0596969593	i = 58	maxMUB	= 0.0439981713	i = 67
maxMUD	= 0.0262863007	i = 73	maxMUD	= 0.0148849328	i = 45
maxMUK	= 0.0285117246	i = 69	maxMUK	= 0.0155243302	i = 14
maxMUN	= 0.0207127359	i = 23	maxMUN	= 0.0121339364	i = 28
maxMUP	= 0.0156522907	i = 79	maxMUP	= 0.0156573876	i = 64
maxMUS	= 0.0151348833	i = 27	maxMUS	= 0.0156119226	i = 49
maxMUBs	= 0.0290334199	i = 66	maxMUBs	= 0.0115209701	i = 85
maxMUDs	= 0.0301568516	i = 62	maxMUDs	= 0.0299700382	i = 46
maxMUKs	= 0.0196872115	i = 22	maxMUKs	= 0.0202375405	i = 57
maxMUNs	= 0.2387821200	i = 32	maxMUNs	= 0.2374949912	i = 28
maxSIGB	= 0.0115401877	i = 85	maxSIGB	= 0.0229623764	i = 38
maxSIGD	= 0.0233297887	i = 36	maxSIGD	= 0.0115996319	i = 71
maxSIGK	= 0.0195671501	i = 31	maxSIGK	= 0.0397582847	i = 32
maxSIGN	= 0.0229002833	i = 72	maxSIGN	= 0.0141798637	i = 34
maxSIGP	= 0.0236156491	i = 33	maxSIGP	= 0.0245648390	i = 54
maxSIGS	= 0.0147282467	i = 52	maxSIGS	= 0.0184902031	i = 36
maxSIGBs	= 0.0166063501	i = 25	maxSIGBs	= 0.0136349146	i = 67
maxSIGDs	= 0.0232533210	i = 48	maxSIGDs	= 0.0092879506	i = 7
maxSIGKs	= 0.0255387755	i = 20	maxSIGKs	= 0.0200653326	i = 40
maxSIGNs	= 0.0138070613	i = 47	maxSIGNs	= 0.0183652644	i = 20
maxSIMon	= 0.6661630426	i = 45	maxSIMon	= 0.6659226275	i = 44

K-S Statistics and Indices for PC, Z-score, OF 4

5000 runs			10000 runs		
maxDCO	= 0.0265181680	i = 58	maxDCO	= 0.0186621057	i = 64
maxDSG	= 0.0267116322	i = 41	maxDSG	= 0.0275161999	i = 54
maxDSO	= 0.0323274500	i = 30	maxDSO	= 0.0258883296	i = 67
maxGT	= 0.0307354747	i = 54	maxGT	= 0.0153433268	i = 19
maxPC	= 0.0295136515	i = 43	maxPC	= 0.0241353098	i = 34
maxSOGCO	= 0.1259152519	i = 53	maxSOGCO	= 0.1276357872	i = 64
maxSUPP	= 0.1869974882	i = 46	maxSUPP	= 0.1950366642	i = 44
maxZPC	= 0.2803843300	i = 40	maxZPC	= 0.2875972226	i = 46
maxIDS	= 0.0651449858	i = 56	maxIDS	= 0.0574547376	i = 19
maxMDS	= 0.0421380871	i = 66	maxMDS	= 0.0303788785	i = 26
maxMUB	= 0.2541463639	i = 38	maxMUB	= 0.2113051701	i = 37
maxMUD	= 0.0376698479	i = 44	maxMUD	= 0.0266865052	i = 71
maxMUK	= 0.0273202237	i = 45	maxMUK	= 0.0365158384	i = 28
maxMUN	= 0.1290984887	i = 53	maxMUN	= 0.1217787913	i = 70
maxMUP	= 0.0330584737	i = 16	maxMUP	= 0.0212225714	i = 29
maxMUS	= 0.0350505845	i = 31	maxMUS	= 0.0228244848	i = 60
maxMUBs	= 0.0397105032	i = 44	maxMUBs	= 0.0421243893	i = 32
maxMUDs	= 0.0540825421	i = 37	maxMUDs	= 0.0240806148	i = 74
maxMUKs	= 0.0675843488	i = 77	maxMUKs	= 0.0760918134	i = 60
maxMUNs	= 0.1507197263	i = 44	maxMUNs	= 0.1502998952	i = 52
maxSIGB	= 0.0925540888	i = 51	maxSIGB	= 0.0899528140	i = 61
maxSIGD	= 0.0486327038	i = 34	maxSIGD	= 0.0102613307	i = 8
maxSIGK	= 0.0636008411	i = 14	maxSIGK	= 0.0440646698	i = 56
maxSIGN	= 0.0476610992	i = 55	maxSIGN	= 0.0248630308	i = 62
maxSIGP	= 0.0283492965	i = 40	maxSIGP	= 0.0152617478	i = 93
maxSIGS	= 0.0253513143	i = 52	maxSIGS	= 0.0378275904	i = 23
maxSIGBs	= 0.1135549558	i = 60	maxSIGBs	= 0.1506030351	i = 45
maxSIGDs	= 0.1285523626	i = 58	maxSIGDs	= 0.0935663895	i = 47
maxSIGKs	= 0.1755838105	i = 43	maxSIGKs	= 0.1599150838	i = 50
maxSIGNs	= 0.1089428677	i = 40	maxSIGNs	= 0.1310176044	i = 32
maxSIMon	= 0.5154683959	i = 44	maxSIMon	= 0.5120477237	i = 45

K-S Statistics and Indices for PC, Z-score, Noise, OF 4

5000 runs			10000 runs		
maxDCO	= 0.0244261295	i = 29	maxDCO	= 0.0118462008	i = 54
maxDSG	= 0.0184196379	i = 48	maxDSG	= 0.0071772963	i = 82
maxDSO	= 0.0328562363	i = 49	maxDSO	= 0.0185128675	i = 20
maxGT	= 0.0250106104	i = 79	maxGT	= 0.0083002746	i = 63
maxPC	= 0.0181593297	i = 65	maxPC	= 0.0161957075	i = 36
maxSOGCO	= 0.0546889842	i = 60	maxSOGCO	= 0.0465995321	i = 53
maxSUPP	= 0.0435959725	i = 30	maxSUPP	= 0.0246302512	i = 47
maxZPC	= 0.1972020906	i = 18	maxZPC	= 0.2053178720	i = 24
maxIDS	= 0.0554537407	i = 43	maxIDS	= 0.0291954023	i = 29
maxMDS	= 0.0304908266	i = 19	maxMDS	= 0.0251286746	i = 79
maxMUB	= 0.0281472439	i = 21	maxMUB	= 0.0291465772	i = 67
maxMUD	= 0.0184754183	i = 83	maxMUD	= 0.0150300071	i = 42
maxMUK	= 0.0269184597	i = 69	maxMUK	= 0.0210009155	i = 35
maxMUN	= 0.0260542686	i = 23	maxMUN	= 0.0159292035	i = 28
maxMUP	= 0.0233469012	i = 17	maxMUP	= 0.0125785780	i = 33
maxMUS	= 0.0102708580	i = 29	maxMUS	= 0.0192757603	i = 54
maxMUBs	= 0.0208472953	i = 66	maxMUBs	= 0.0149689757	i = 57
maxMUDs	= 0.0232078545	i = 13	maxMUDs	= 0.0282433120	i = 46
maxMUKs	= 0.0181892409	i = 68	maxMUKs	= 0.0091587834	i = 26
maxMUNs	= 0.0258893528	i = 55	maxMUNs	= 0.0161428135	i = 46
maxSIGB	= 0.0221294346	i = 85	maxSIGB	= 0.0434625165	i = 38
maxSIGD	= 0.0246169144	i = 21	maxSIGD	= 0.0121350829	i = 87
maxSIGK	= 0.0343154176	i = 58	maxSIGK	= 0.0334330180	i = 32
maxSIGN	= 0.0286355240	i = 72	maxSIGN	= 0.0168873970	i = 33
maxSIGP	= 0.0202870666	i = 13	maxSIGP	= 0.0230393653	i = 59
maxSIGS	= 0.0285983371	i = 67	maxSIGS	= 0.0159373411	i = 35
maxSIGBs	= 0.0233743871	i = 43	maxSIGBs	= 0.0115206998	i = 60
maxSIGDs	= 0.0244091528	i = 48	maxSIGDs	= 0.0116488658	i = 38
maxSIGKs	= 0.0224916026	i = 53	maxSIGKs	= 0.0135672872	i = 40
maxSIGNs	= 0.0143897105	i = 22	maxSIGNs	= 0.0236211983	i = 62
maxSIMon	= 0.7995157479	i = 44	maxSIMon	= 0.7874926254	i = 44

A.1.3 Pearson's Correlation with Z-score and Mutual Information

K-S Statistics and Indices for PC, Z-score, MI, OF 1

5000 runs			10000 runs		
maxDCO	= 0.0195621913	i = 3	maxDCO	= 0.0439745934	i = 67
maxDSG	= 0.0288222854	i = 77	maxDSG	= 0.0287157886	i = 55
maxDSO	= 0.0509805443	i = 68	maxDSO	= 0.0452270961	i = 32
maxGT	= 0.0228213518	i = 83	maxGT	= 0.0230372676	i = 31
maxPC	= 0.0218109246	i = 24	maxPC	= 0.0341830621	i = 29
maxSOGCO	= 0.2036322882	i = 55	maxSOGCO	= 0.1815431628	i = 64
maxSUPP	= 0.1841541723	i = 47	maxSUPP	= 0.1540058179	i = 49
maxZPC	= 0.2488568971	i = 48	maxZPC	= 0.2738756850	i = 45
maxZMI	= 0.0361600569	i = 31	maxZMI	= 0.0206382628	i = 56
maxDIM	= 0.0275421319	i = 0	maxDIM	= 0.0357811738	i = 14
maxMI	= 0.0544394316	i = 53	maxMI	= 0.0383842650	i = 53
maxNEG	= 0.0308861287	i = 8	maxNEG	= 0.0297672199	i = 38
maxPEAK	= 0.0257099716	i = 11	maxPEAK	= 0.0212445861	i = 1
maxSFACT	= 0.0320586200	i = 20	maxSFACT	= 0.0234435560	i = 81
maxIDS	= 0.1587382746	i = 23	maxIDS	= 0.1468557264	i = 33
maxMDS	= 0.0529656382	i = 36	maxMDS	= 0.0267738582	i = 49
maxMUB	= 0.2433440577	i = 32	maxMUB	= 0.2672470221	i = 37
maxMUD	= 0.0289421215	i = 43	maxMUD	= 0.0327706635	i = 45
maxMUK	= 0.0392035125	i = 12	maxMUK	= 0.0237260357	i = 41
maxMUN	= 0.1280526335	i = 57	maxMUN	= 0.1217153134	i = 55
maxMUP	= 0.0600861831	i = 56	maxMUP	= 0.0347255024	i = 75
maxMUS	= 0.0370848869	i = 75	maxMUS	= 0.0198660134	i = 34
maxMUBs	= 0.0508401649	i = 24	maxMUBs	= 0.0319506431	i = 61
maxMUDs	= 0.0344001787	i = 31	maxMUDs	= 0.0188522711	i = 57
maxMUKs	= 0.0606861242	i = 44	maxMUKs	= 0.0621543027	i = 48
maxMUNs	= 0.1502253489	i = 59	maxMUNs	= 0.1532029460	i = 42
maxSIGB	= 0.1152503375	i = 51	maxSIGB	= 0.0789306588	i = 61
maxSIGD	= 0.0394222608	i = 42	maxSIGD	= 0.0309356759	i = 54
maxSIGK	= 0.0446128743	i = 58	maxSIGK	= 0.0788345516	i = 62
maxSIGN	= 0.0231306430	i = 71	maxSIGN	= 0.0236260654	i = 77
maxSIGP	= 0.0231043932	i = 80	maxSIGP	= 0.0155013341	i = 35
maxSIGS	= 0.0413559964	i = 68	maxSIGS	= 0.0223858117	i = 66
maxSIGBs	= 0.1549640092	i = 31	maxSIGBs	= 0.1623038248	i = 36
maxSIGDs	= 0.1916829206	i = 65	maxSIGDs	= 0.1670680084	i = 64
maxSIGKs	= 0.1965319063	i = 45	maxSIGKs	= 0.1702948540	i = 37
maxSIGNs	= 0.1348106267	i = 45	maxSIGNs	= 0.1308100676	i = 46
maxSIMon	= 0.5006526310	i = 45	maxSIMon	= 0.4970192170	i = 44

K-S Statistics and Indices for PC, Z-score, MI, Noise, OF 1

5000 runs			10000 runs		
maxDCO	= 0.0191904731	i = 14	maxDCO	= 0.0147903893	i = 66
maxDSG	= 0.0229582232	i = 64	maxDSG	= 0.0094509740	i = 65
maxDSO	= 0.0274195795	i = 59	maxDSO	= 0.0128015645	i = 83
maxGT	= 0.0251576294	i = 81	maxGT	= 0.0125231702	i = 45
maxPC	= 0.0332516337	i = 23	maxPC	= 0.0157911208	i = 82
maxSOGCO	= 0.0292475867	i = 60	maxSOGCO	= 0.0323162308	i = 62
maxSUPP	= 0.0571230699	i = 39	maxSUPP	= 0.0545565346	i = 29
maxZPC	= 0.0786260906	i = 14	maxZPC	= 0.0838574401	i = 14
maxZMI	= 0.0116099973	i = 80	maxZMI	= 0.0143922927	i = 39
maxDIM	= 0.0088921434	i = 28	maxDIM	= 0.0066374980	i = 57
maxMI	= 0.0154207904	i = 57	maxMI	= 0.0183697184	i = 50
maxNEG	= 0.0134200845	i = 11	maxNEG	= 0.0208151690	i = 54
maxPEAK	= 0.0186249416	i = 1	maxPEAK	= 0.0112477194	i = 11
maxSFACT	= 0.0406947235	i = 32	maxSFACT	= 0.0204785145	i = 46
maxIDS	= 0.0427491892	i = 33	maxIDS	= 0.0398252121	i = 36
maxMDS	= 0.0284336360	i = 16	maxMDS	= 0.0253822330	i = 53
maxMUB	= 0.1007380370	i = 43	maxMUB	= 0.0943179742	i = 59
maxMUD	= 0.0258668715	i = 70	maxMUD	= 0.0168413734	i = 39
maxMUK	= 0.0320348363	i = 51	maxMUK	= 0.0135161037	i = 40
maxMUN	= 0.0248779381	i = 85	maxMUN	= 0.0214302626	i = 32
maxMUP	= 0.0310032114	i = 39	maxMUP	= 0.0341650519	i = 40
maxMUS	= 0.0341562381	i = 58	maxMUS	= 0.0180642107	i = 20
maxMUBs	= 0.0143115124	i = 44	maxMUBs	= 0.0150008880	i = 86
maxMUDs	= 0.0100469238	i = 59	maxMUDs	= 0.0191383949	i = 13
maxMUKs	= 0.0352760572	i = 56	maxMUKs	= 0.0288876612	i = 60
maxMUNs	= 0.2414696369	i = 34	maxMUNs	= 0.2251103460	i = 25
maxSIGB	= 0.0350942227	i = 75	maxSIGB	= 0.0163112053	i = 48
maxSIGD	= 0.0144501283	i = 64	maxSIGD	= 0.0139861293	i = 81
maxSIGK	= 0.0320054968	i = 38	maxSIGK	= 0.0319444857	i = 45
maxSIGN	= 0.0163018529	i = 30	maxSIGN	= 0.0271456805	i = 84
maxSIGP	= 0.0191237126	i = 53	maxSIGP	= 0.0163925904	i = 72
maxSIGS	= 0.0197733974	i = 66	maxSIGS	= 0.0173020325	i = 65
maxSIGBs	= 0.0285383445	i = 48	maxSIGBs	= 0.0194988016	i = 32
maxSIGDs	= 0.0285831443	i = 74	maxSIGDs	= 0.0146165050	i = 15
maxSIGKs	= 0.0275025031	i = 82	maxSIGKs	= 0.0234072113	i = 44
maxSIGNs	= 0.0163842494	i = 79	maxSIGNs	= 0.0138175332	i = 50
maxSIMon	= 0.6718650820	i = 45	maxSIMon	= 0.6745735654	i = 44

K-S Statistics and Indices for PC, Z-score, MI, OF 2/3

5000 runs			10000 runs		
maxDCO	= 0.0225817923	i = 6	maxDCO	= 0.0473955280	i = 68
maxDSG	= 0.0243307338	i = 80	maxDSG	= 0.0287539191	i = 55
maxDSO	= 0.0444042144	i = 68	maxDSO	= 0.0423243063	i = 32
maxGT	= 0.0258594029	i = 85	maxGT	= 0.0247866986	i = 31
maxPC	= 0.0211358633	i = 24	maxPC	= 0.0310044574	i = 13
maxSOGCO	= 0.1884043262	i = 64	maxSOGCO	= 0.1666525646	i = 64
maxSUPP	= 0.1860504626	i = 47	maxSUPP	= 0.1612224070	i = 39
maxZPC	= 0.2435129014	i = 48	maxZPC	= 0.2749406956	i = 45
maxZMI	= 0.0348059966	i = 31	maxZMI	= 0.0196994915	i = 57
maxDIM	= 0.0231749467	i = 0	maxDIM	= 0.0333578902	i = 14
maxMI	= 0.0618564890	i = 60	maxMI	= 0.0408749658	i = 53
maxNEG	= 0.0303347223	i = 8	maxNEG	= 0.0271324143	i = 58
maxPEAK	= 0.0247606145	i = 11	maxPEAK	= 0.0247799002	i = 1
maxSFACT	= 0.0366693033	i = 20	maxSFACT	= 0.0252357590	i = 81
maxIDS	= 0.1623154574	i = 23	maxIDS	= 0.1453472781	i = 33
maxMDS	= 0.0461230018	i = 36	maxMDS	= 0.0238949169	i = 49
maxMUB	= 0.2461006436	i = 32	maxMUB	= 0.2618493184	i = 37
maxMUD	= 0.0309856109	i = 43	maxMUD	= 0.0300706912	i = 49
maxMUK	= 0.0325995942	i = 12	maxMUK	= 0.0193838263	i = 41
maxMUN	= 0.1240891695	i = 58	maxMUN	= 0.1260701128	i = 55
maxMUP	= 0.0614872843	i = 56	maxMUP	= 0.0333398837	i = 75
maxMUS	= 0.0378887083	i = 75	maxMUS	= 0.0266643368	i = 42
maxMUBs	= 0.0510171699	i = 24	maxMUBs	= 0.0289255321	i = 61
maxMUDs	= 0.0289156378	i = 31	maxMUDs	= 0.0180834990	i = 35
maxMUKs	= 0.0672831355	i = 44	maxMUKs	= 0.0642076386	i = 48
maxMUNs	= 0.1475469035	i = 59	maxMUNs	= 0.1517814825	i = 42
maxSIGB	= 0.1050902069	i = 51	maxSIGB	= 0.0797750031	i = 61
maxSIGD	= 0.0403223701	i = 48	maxSIGD	= 0.0280593822	i = 54
maxSIGK	= 0.0381402382	i = 58	maxSIGK	= 0.0755508224	i = 62
maxSIGN	= 0.0212524349	i = 71	maxSIGN	= 0.0257881730	i = 77
maxSIGP	= 0.0234158123	i = 80	maxSIGP	= 0.0151453139	i = 63
maxSIGS	= 0.0392272591	i = 67	maxSIGS	= 0.0245123749	i = 66
maxSIGBs	= 0.1579199363	i = 31	maxSIGBs	= 0.1596206542	i = 44
maxSIGDs	= 0.1869587649	i = 65	maxSIGDs	= 0.1651989063	i = 64
maxSIGKs	= 0.1822436832	i = 45	maxSIGKs	= 0.1718321929	i = 37
maxSIGNs	= 0.1379979620	i = 45	maxSIGNs	= 0.1303766090	i = 46
maxSIMon	= 0.5028427655	i = 44	maxSIMon	= 0.4983081688	i = 44

K-S Statistics and Indices for PC, Z-score, MI, Noise, OF 2/3

5000 runs			10000 runs		
maxDCO	= 0.0174002574	i = 14	maxDCO	= 0.0198564901	i = 22
maxDSG	= 0.0211239811	i = 64	maxDSG	= 0.0107642890	i = 79
maxDSO	= 0.0305448305	i = 56	maxDSO	= 0.0124011942	i = 83
maxGT	= 0.0258515659	i = 81	maxGT	= 0.0121843370	i = 45
maxPC	= 0.0289317889	i = 23	maxPC	= 0.0161513223	i = 82
maxSOGCO	= 0.0337451737	i = 60	maxSOGCO	= 0.0302610964	i = 68
maxSUPP	= 0.0406778207	i = 34	maxSUPP	= 0.0418566714	i = 30
maxZPC	= 0.0757700558	i = 14	maxZPC	= 0.0876661852	i = 14
maxZMI	= 0.0179322179	i = 31	maxZMI	= 0.0116937892	i = 21
maxDIM	= 0.0138138138	i = 14	maxDIM	= 0.0082347247	i = 85
maxMI	= 0.0238095238	i = 57	maxMI	= 0.0219364511	i = 50
maxNEG	= 0.0113513514	i = 11	maxNEG	= 0.0171103306	i = 54
maxPEAK	= 0.0221879022	i = 11	maxPEAK	= 0.0166331399	i = 4
maxSFACT	= 0.0387473187	i = 34	maxSFACT	= 0.0235196658	i = 46
maxIDS	= 0.0438610039	i = 33	maxIDS	= 0.0400084962	i = 36
maxMDS	= 0.0229601030	i = 16	maxMDS	= 0.0210777524	i = 26
maxMUB	= 0.0807979408	i = 40	maxMUB	= 0.0668134090	i = 59
maxMUD	= 0.0196138996	i = 70	maxMUD	= 0.0123451466	i = 41
maxMUK	= 0.0325868726	i = 42	maxMUK	= 0.0119304639	i = 40
maxMUN	= 0.0290948091	i = 41	maxMUN	= 0.0146952187	i = 48
maxMUP	= 0.0311454311	i = 39	maxMUP	= 0.0307682315	i = 40
maxMUS	= 0.0282539683	i = 58	maxMUS	= 0.0160026477	i = 20
maxMUBs	= 0.0135306735	i = 79	maxMUBs	= 0.0147823457	i = 86
maxMUDs	= 0.0138652939	i = 69	maxMUDs	= 0.0174896561	i = 13
maxMUKs	= 0.0355298155	i = 57	maxMUKs	= 0.0288578103	i = 61
maxMUNs	= 0.2397854998	i = 34	maxMUNs	= 0.2314334349	i = 32
maxSIGB	= 0.0353839554	i = 75	maxSIGB	= 0.0250984282	i = 48
maxSIGD	= 0.0210210210	i = 64	maxSIGD	= 0.0144985282	i = 81
maxSIGK	= 0.0241613042	i = 26	maxSIGK	= 0.0250090314	i = 39
maxSIGN	= 0.0181038181	i = 30	maxSIGN	= 0.0253209600	i = 84
maxSIGP	= 0.0223680824	i = 46	maxSIGP	= 0.0141893059	i = 66
maxSIGS	= 0.0201630202	i = 66	maxSIGS	= 0.0190912915	i = 66
maxSIGBs	= 0.0267610468	i = 40	maxSIGBs	= 0.0164240526	i = 32
maxSIGDs	= 0.0217245817	i = 39	maxSIGDs	= 0.0144953853	i = 78
maxSIGKs	= 0.0258344058	i = 87	maxSIGKs	= 0.0248248249	i = 44
maxSIGNs	= 0.0178464178	i = 78	maxSIGNs	= 0.0142511154	i = 50
maxSIMon	= 0.6936936937	i = 45	maxSIMon	= 0.6952723208	i = 44

K-S Statistics and Indices for PC, Z-score, MI, OF 4

5000 runs			10000 runs		
maxDCO	= 0.0250218613	i = 89	maxDCO	= 0.0395712269	i = 63
maxDSG	= 0.0367439556	i = 77	maxDSG	= 0.0246971339	i = 55
maxDSO	= 0.0320677968	i = 68	maxDSO	= 0.0346088184	i = 29
maxGT	= 0.0265786202	i = 42	maxGT	= 0.0189467809	i = 25
maxPC	= 0.0178857357	i = 68	maxPC	= 0.0301552907	i = 29
maxSOGCO	= 0.1755588843	i = 56	maxSOGCO	= 0.1559653961	i = 63
maxSUPP	= 0.1743458933	i = 46	maxSUPP	= 0.1607258938	i = 54
maxZPC	= 0.2506911304	i = 48	maxZPC	= 0.2711464592	i = 40
maxZMI	= 0.0388059097	i = 35	maxZMI	= 0.0221872025	i = 57
maxDIM	= 0.0238709576	i = 57	maxDIM	= 0.0280754457	i = 14
maxMI	= 0.0719030496	i = 53	maxMI	= 0.0327226764	i = 53
maxNEG	= 0.0281987737	i = 8	maxNEG	= 0.0281887127	i = 38
maxPEAK	= 0.0098431346	i = 11	maxPEAK	= 0.0174267048	i = 1
maxSFACT	= 0.0390751728	i = 20	maxSFACT	= 0.0187629272	i = 81
maxIDS	= 0.1525375432	i = 23	maxIDS	= 0.1411832299	i = 33
maxMDS	= 0.0259623215	i = 36	maxMDS	= 0.0134853410	i = 49
maxMUB	= 0.2279665016	i = 32	maxMUB	= 0.2397468728	i = 37
maxMUD	= 0.0246663294	i = 84	maxMUD	= 0.0268541318	i = 51
maxMUK	= 0.0356512180	i = 12	maxMUK	= 0.0176039923	i = 41
maxMUN	= 0.1175941212	i = 58	maxMUN	= 0.1229505237	i = 55
maxMUP	= 0.0606508586	i = 56	maxMUP	= 0.0303489937	i = 74
maxMUS	= 0.0355930520	i = 75	maxMUS	= 0.0293394399	i = 34
maxMUBs	= 0.0418802145	i = 28	maxMUBs	= 0.0311254473	i = 69
maxMUDs	= 0.0315952793	i = 31	maxMUDs	= 0.0171476411	i = 57
maxMUKs	= 0.0644564872	i = 42	maxMUKs	= 0.0550067304	i = 48
maxMUNs	= 0.1358105538	i = 57	maxMUNs	= 0.1328293772	i = 54
maxSIGB	= 0.1075915200	i = 51	maxSIGB	= 0.0876982173	i = 53
maxSIGD	= 0.0352898042	i = 40	maxSIGD	= 0.0305853771	i = 61
maxSIGK	= 0.0361341924	i = 58	maxSIGK	= 0.0673659674	i = 45
maxSIGN	= 0.0218161926	i = 71	maxSIGN	= 0.0299894941	i = 49
maxSIGP	= 0.0225226815	i = 24	maxSIGP	= 0.0154272957	i = 21
maxSIGS	= 0.0430350395	i = 68	maxSIGS	= 0.0182918021	i = 28
maxSIGBs	= 0.1331917741	i = 31	maxSIGBs	= 0.1509356840	i = 35
maxSIGDs	= 0.1672019251	i = 65	maxSIGDs	= 0.1496339341	i = 64
maxSIGKs	= 0.1694409915	i = 45	maxSIGKs	= 0.1702058505	i = 48
maxSIGNs	= 0.1390900737	i = 45	maxSIGNs	= 0.1245461112	i = 46
maxSIMon	= 0.5139984132	i = 44	maxSIMon	= 0.5085853114	i = 44

K-S Statistics and Indices for PC, Z-score, MI, Noise, OF 4

5000 runs			10000 runs		
maxDCO	= 0.0231882045	i = 14	maxDCO	= 0.0169577833	i = 68
maxDSG	= 0.0183735628	i = 19	maxDSG	= 0.0158947882	i = 67
maxDSO	= 0.0236128651	i = 45	maxDSO	= 0.0137405397	i = 82
maxGT	= 0.0267332043	i = 81	maxGT	= 0.0134449583	i = 43
maxPC	= 0.0319376273	i = 69	maxPC	= 0.0203624195	i = 75
maxSOGCO	= 0.0333356961	i = 56	maxSOGCO	= 0.0345809275	i = 68
maxSUPP	= 0.0311644943	i = 39	maxSUPP	= 0.0317103725	i = 30
maxZPC	= 0.1143513583	i = 30	maxZPC	= 0.1312668147	i = 20
maxZMI	= 0.0181539994	i = 40	maxZMI	= 0.0128885935	i = 43
maxDIM	= 0.0121160072	i = 42	maxDIM	= 0.0152245829	i = 28
maxMI	= 0.0267597255	i = 33	maxMI	= 0.0211245045	i = 76
maxNEG	= 0.0122764203	i = 13	maxNEG	= 0.0169388099	i = 66
maxPEAK	= 0.0148439935	i = 1	maxPEAK	= 0.0152453326	i = 4
maxSFACT	= 0.0349931905	i = 34	maxSFACT	= 0.0142461204	i = 47
maxIDS	= 0.0447817619	i = 49	maxIDS	= 0.0350359663	i = 36
maxMDS	= 0.0211812733	i = 43	maxMDS	= 0.0193406007	i = 26
maxMUB	= 0.0617112858	i = 44	maxMUB	= 0.0416995875	i = 59
maxMUD	= 0.0201429283	i = 70	maxMUD	= 0.0162044987	i = 46
maxMUK	= 0.0169779050	i = 79	maxMUK	= 0.0155451124	i = 24
maxMUN	= 0.0247404178	i = 42	maxMUN	= 0.0132723407	i = 32
maxMUP	= 0.0224806512	i = 83	maxMUP	= 0.0320072457	i = 49
maxMUS	= 0.0312241268	i = 58	maxMUS	= 0.0132137250	i = 17
maxMUBs	= 0.0210840290	i = 35	maxMUBs	= 0.0141468128	i = 42
maxMUDs	= 0.0097708906	i = 70	maxMUDs	= 0.0102737354	i = 13
maxMUKs	= 0.0217570244	i = 88	maxMUKs	= 0.0193309929	i = 61
maxMUNs	= 0.0291450258	i = 39	maxMUNs	= 0.0233819359	i = 65
maxSIGB	= 0.0224158752	i = 42	maxSIGB	= 0.0117686782	i = 14
maxSIGD	= 0.0354990220	i = 63	maxSIGD	= 0.0112873189	i = 4
maxSIGK	= 0.0288677587	i = 39	maxSIGK	= 0.0326504845	i = 47
maxSIGN	= 0.0212521573	i = 59	maxSIGN	= 0.0201792252	i = 77
maxSIGP	= 0.0167170328	i = 53	maxSIGP	= 0.0139253488	i = 66
maxSIGS	= 0.0161075275	i = 80	maxSIGS	= 0.0205418191	i = 66
maxSIGBs	= 0.0258589693	i = 48	maxSIGBs	= 0.0111451394	i = 9
maxSIGDs	= 0.0221435909	i = 74	maxSIGDs	= 0.0145375035	i = 16
maxSIGKs	= 0.0209908029	i = 87	maxSIGKs	= 0.0285473848	i = 50
maxSIGNs	= 0.0152284705	i = 94	maxSIGNs	= 0.0093580062	i = 69
maxSIMon	= 0.8172948413	i = 45	maxSIMon	= 0.8175049117	i = 44

A.1.4 Pearson's Correlation with Z-score, Noise, and Relaxed Module

K-S Statistics and Indices for PC, Z-score, Noise, RM, OF 1

5000 runs			10000 runs		
maxIDS	= 0.0664727198	i = 59	maxIDS	= 0.0727332687	i = 43
maxMDS	= 0.0201041188	i = 53	maxMDS	= 0.0222241432	i = 16
maxMUB	= 0.1059323908	i = 48	maxMUB	= 0.1035736431	i = 37
maxMUD	= 0.0181850661	i = 28	maxMUD	= 0.0249665154	i = 49
maxMUK	= 0.0130114497	i = 87	maxMUK	= 0.0381842457	i = 38
maxMUN	= 0.0141496113	i = 94	maxMUN	= 0.0419974835	i = 61
maxMUP	= 0.0163391688	i = 64	maxMUP	= 0.0236708897	i = 25
maxMUS	= 0.0327446877	i = 42	maxMUS	= 0.0257605398	i = 52
maxMUBs	= 0.0263784685	i = 51	maxMUBs	= 0.0198147490	i = 31
maxMUDs	= 0.0182582215	i = 57	maxMUDs	= 0.0242909544	i = 53
maxMUKs	= 0.0310893346	i = 54	maxMUKs	= 0.0257054703	i = 20
maxMUNs	= 0.3184981031	i = 27	maxMUNs	= 0.2945296633	i = 26
maxSIGB	= 0.0369638817	i = 70	maxSIGB	= 0.0323242608	i = 71
maxSIGD	= 0.0237312646	i = 12	maxSIGD	= 0.0270819926	i = 46
maxSIGK	= 0.0384423008	i = 40	maxSIGK	= 0.0238832699	i = 70
maxSIGN	= 0.0129757226	i = 88	maxSIGN	= 0.0294647059	i = 54
maxSIGP	= 0.0262117423	i = 60	maxSIGP	= 0.0175233912	i = 63
maxSIGS	= 0.0181102094	i = 83	maxSIGS	= 0.0144924381	i = 37
maxSIGBs	= 0.0172646694	i = 50	maxSIGBs	= 0.0254371670	i = 83
maxSIGDs	= 0.0170877354	i = 15	maxSIGDs	= 0.0181724847	i = 76
maxSIGKs	= 0.0457782541	i = 63	maxSIGKs	= 0.0486231042	i = 54
maxSIGNs	= 0.0182241957	i = 52	maxSIGNs	= 0.0168915869	i = 65
maxSIMon	= 0.7064597901	i = 45	maxSIMon	= 0.7116637264	i = 45

K-S Statistics and Indices for PC, Z-score, Noise, RM, OF 2/3

5000 runs			10000 runs		
maxIDS	= 0.0688063515	i = 59	maxIDS	= 0.0733185646	i = 43
maxMDS	= 0.0212640269	i = 53	maxMDS	= 0.0220782863	i = 16
maxMUB	= 0.1041871759	i = 48	maxMUB	= 0.0992522112	i = 38
maxMUD	= 0.0201946601	i = 28	maxMUD	= 0.0247143371	i = 49
maxMUK	= 0.0122182296	i = 87	maxMUK	= 0.0384664612	i = 38
maxMUN	= 0.0149021774	i = 88	maxMUN	= 0.0438175352	i = 61
maxMUP	= 0.0169726320	i = 64	maxMUP	= 0.0244726537	i = 25
maxMUS	= 0.0330277417	i = 42	maxMUS	= 0.0241731025	i = 52
maxMUBs	= 0.0246976865	i = 51	maxMUBs	= 0.0162697159	i = 31
maxMUDs	= 0.0197404849	i = 57	maxMUDs	= 0.0241628905	i = 53
maxMUKs	= 0.0322488160	i = 54	maxMUKs	= 0.0270753453	i = 20
maxMUNs	= 0.3164653281	i = 27	maxMUNs	= 0.2923551806	i = 26
maxSIGB	= 0.0388637400	i = 70	maxSIGB	= 0.0342244074	i = 71
maxSIGD	= 0.0227967408	i = 12	maxSIGD	= 0.0272700536	i = 46
maxSIGK	= 0.0369373170	i = 40	maxSIGK	= 0.0250206963	i = 70
maxSIGN	= 0.0136921759	i = 88	maxSIGN	= 0.0284832360	i = 54
maxSIGP	= 0.0269335736	i = 60	maxSIGP	= 0.0193060760	i = 63
maxSIGS	= 0.0170837129	i = 84	maxSIGS	= 0.0163030750	i = 42
maxSIGBs	= 0.0162945963	i = 50	maxSIGBs	= 0.0240546436	i = 52
maxSIGDs	= 0.0168163712	i = 15	maxSIGDs	= 0.0195266546	i = 76
maxSIGKs	= 0.0447136658	i = 63	maxSIGKs	= 0.0458252091	i = 54
maxSIGNs	= 0.0179332957	i = 52	maxSIGNs	= 0.0179002277	i = 65
maxSIMon	= 0.7083536302	i = 45	maxSIMon	= 0.7154161311	i = 45

K-S Statistics and Indices for PC, Z-score, Noise, RM, OF 4

5000 runs			10000 runs		
maxIDS	= 0.0310838029	i = 59	maxIDS	= 0.0280352317	i = 46
maxMDS	= 0.0177996538	i = 53	maxMDS	= 0.0108898008	i = 16
maxMUB	= 0.0472698377	i = 47	maxMUB	= 0.0364167751	i = 53
maxMUD	= 0.0303955947	i = 29	maxMUD	= 0.0254649184	i = 31
maxMUK	= 0.0170579148	i = 57	maxMUK	= 0.0177579822	i = 38
maxMUN	= 0.0131461548	i = 91	maxMUN	= 0.0240616555	i = 63
maxMUP	= 0.0272622761	i = 65	maxMUP	= 0.0224321890	i = 25
maxMUS	= 0.0297932923	i = 62	maxMUS	= 0.0222920629	i = 50
maxMUBs	= 0.0318902919	i = 31	maxMUBs	= 0.0262316084	i = 31
maxMUDs	= 0.0365042037	i = 57	maxMUDs	= 0.0313742368	i = 60
maxMUKs	= 0.0244289867	i = 33	maxMUKs	= 0.0225002502	i = 16
maxMUNs	= 0.0354074620	i = 81	maxMUNs	= 0.0125292763	i = 14
maxSIGB	= 0.0322957801	i = 42	maxSIGB	= 0.0219097187	i = 19
maxSIGD	= 0.0350191229	i = 19	maxSIGD	= 0.0194835352	i = 56
maxSIGK	= 0.0404834382	i = 40	maxSIGK	= 0.0134481033	i = 53
maxSIGN	= 0.0354473698	i = 51	maxSIGN	= 0.0195696127	i = 31
maxSIGP	= 0.0197116994	i = 60	maxSIGP	= 0.0163166850	i = 79
maxSIGS	= 0.0281141220	i = 83	maxSIGS	= 0.0237513762	i = 51
maxSIGBs	= 0.0220492673	i = 53	maxSIGBs	= 0.0296566910	i = 74
maxSIGDs	= 0.0203846818	i = 15	maxSIGDs	= 0.0197898108	i = 76
maxSIGKs	= 0.0384066330	i = 53	maxSIGKs	= 0.0461475328	i = 54
maxSIGNs	= 0.0156324552	i = 46	maxSIGNs	= 0.0105715144	i = 68
maxSIMon	= 0.9130298133	i = 45	maxSIMon	= 0.9171093985	i = 44

A.1.5 Pearson's Correlation with Z-score, MI, Noise, and Relaxed Module

K-S Statistics and Indices for PC, Z-score, MI, Noise, RM, OF 1

5000 runs			10000 runs		
maxIDS	= 0.0634702709	i = 29	maxIDS	= 0.0675120970	i = 43
maxMDS	= 0.0402135494	i = 26	maxMDS	= 0.0233322453	i = 76
maxMUB	= 0.0730315272	i = 48	maxMUB	= 0.0819685486	i = 54
maxMUD	= 0.0273922994	i = 74	maxMUD	= 0.0110620768	i = 8
maxMUK	= 0.0442584199	i = 56	maxMUK	= 0.0233408341	i = 40
maxMUN	= 0.0213809585	i = 18	maxMUN	= 0.0285447740	i = 39
maxMUP	= 0.0224345348	i = 14	maxMUP	= 0.0194517085	i = 39
maxMUS	= 0.0175496786	i = 45	maxMUS	= 0.0102325316	i = 17
maxMUBs	= 0.0290212694	i = 74	maxMUBs	= 0.0156090135	i = 21
maxMUDs	= 0.0399208894	i = 75	maxMUDs	= 0.0118328608	i = 63
maxMUKs	= 0.0181521134	i = 17	maxMUKs	= 0.0271879085	i = 51
maxMUNs	= 0.3202786398	i = 29	maxMUNs	= 0.3107438095	i = 30
maxSIGB	= 0.0374355035	i = 52	maxSIGB	= 0.0467799296	i = 57
maxSIGD	= 0.0240984187	i = 63	maxSIGD	= 0.0163380864	i = 34
maxSIGK	= 0.0439568602	i = 26	maxSIGK	= 0.0263455651	i = 66
maxSIGN	= 0.0420064775	i = 61	maxSIGN	= 0.0407600505	i = 45
maxSIGP	= 0.0243233048	i = 34	maxSIGP	= 0.0204089382	i = 61
maxSIGS	= 0.0087489962	i = 24	maxSIGS	= 0.0142509149	i = 27
maxSIGBs	= 0.0296291808	i = 34	maxSIGBs	= 0.0198285967	i = 80
maxSIGDs	= 0.0152905481	i = 18	maxSIGDs	= 0.0114543144	i = 22
maxSIGKs	= 0.0321974015	i = 47	maxSIGKs	= 0.0113814794	i = 69
maxSIGNs	= 0.0095390073	i = 92	maxSIGNs	= 0.0141522284	i = 32
maxSIMon	= 0.7158185001	i = 45	maxSIMon	= 0.7151422105	i = 45

K-S Statistics and Indices for PC, Z-score, MI, Noise, RM, OF 2/3

5000 runs		10000 runs	
maxIDS = 0.0662208777	i = 29	maxIDS = 0.0695687992	i = 43
maxMDS = 0.0384265356	i = 26	maxMDS = 0.0211149381	i = 76
maxMUB = 0.0691021815	i = 48	maxMUB = 0.0750854976	i = 29
maxMUD = 0.0262149708	i = 74	maxMUD = 0.0112976131	i = 73
maxMUK = 0.0436876392	i = 56	maxMUK = 0.0206176697	i = 40
maxMUN = 0.0254977569	i = 26	maxMUN = 0.0343894303	i = 39
maxMUP = 0.0246195705	i = 14	maxMUP = 0.0197918247	i = 39
maxMUS = 0.0166154562	i = 45	maxMUS = 0.0094772313	i = 17
maxMUBs = 0.0269534999	i = 74	maxMUBs = 0.0181265853	i = 21
maxMUDs = 0.0388195880	i = 75	maxMUDs = 0.0125794004	i = 63
maxMUKs = 0.0163323903	i = 17	maxMUKs = 0.0238817558	i = 59
maxMUNs = 0.3202708088	i = 29	maxMUNs = 0.3127730234	i = 30
maxSIGB = 0.0398684664	i = 52	maxSIGB = 0.0526257669	i = 57
maxSIGD = 0.0245781338	i = 37	maxSIGD = 0.0174165890	i = 33
maxSIGK = 0.0400335314	i = 26	maxSIGK = 0.0221887398	i = 66
maxSIGN = 0.0388770537	i = 61	maxSIGN = 0.0390945108	i = 45
maxSIGP = 0.0259261072	i = 34	maxSIGP = 0.0200993995	i = 61
maxSIGS = 0.0106644221	i = 24	maxSIGS = 0.0122982472	i = 27
maxSIGBs = 0.0285707270	i = 34	maxSIGBs = 0.0228147289	i = 80
maxSIGDs = 0.0154023654	i = 72	maxSIGDs = 0.0099040421	i = 22
maxSIGKs = 0.0312184162	i = 47	maxSIGKs = 0.0099507880	i = 86
maxSIGNs = 0.0130844650	i = 45	maxSIGNs = 0.0138950615	i = 32
maxSIMon = 0.7198965820	i = 45	maxSIMon = 0.7197370846	i = 45

K-S Statistics and Indices for PC, Z-score, MI, Noise, RM, OF 4

5000 runs			10000 runs		
maxIDS	= 0.0257482430	i = 23	maxIDS	= 0.0390756972	i = 36
maxMDS	= 0.0212194335	i = 6	maxMDS	= 0.0145245847	i = 46
maxMUB	= 0.0194622388	i = 61	maxMUB	= 0.0199328539	i = 53
maxMUD	= 0.0193086165	i = 14	maxMUD	= 0.0084914918	i = 9
maxMUK	= 0.0257213029	i = 56	maxMUK	= 0.0147646671	i = 53
maxMUN	= 0.0217963989	i = 82	maxMUN	= 0.0112782600	i = 49
maxMUP	= 0.0276257057	i = 14	maxMUP	= 0.0269478469	i = 41
maxMUS	= 0.0223547628	i = 60	maxMUS	= 0.0149664197	i = 64
maxMUBs	= 0.0274082612	i = 68	maxMUBs	= 0.0208989571	i = 80
maxMUDs	= 0.0267770951	i = 75	maxMUDs	= 0.0151556631	i = 34
maxMUKs	= 0.0139876422	i = 46	maxMUKs	= 0.0132672393	i = 89
maxMUNs	= 0.0276465521	i = 66	maxMUNs	= 0.0242763889	i = 50
maxSIGB	= 0.0262145618	i = 33	maxSIGB	= 0.0249393018	i = 73
maxSIGD	= 0.0261379111	i = 61	maxSIGD	= 0.0127881971	i = 12
maxSIGK	= 0.0388471853	i = 15	maxSIGK	= 0.0231616175	i = 65
maxSIGN	= 0.0243425513	i = 33	maxSIGN	= 0.0211176293	i = 39
maxSIGP	= 0.0381137113	i = 34	maxSIGP	= 0.0080960054	i = 60
maxSIGS	= 0.0169003713	i = 71	maxSIGS	= 0.0201688467	i = 28
maxSIGBs	= 0.0178942656	i = 84	maxSIGBs	= 0.0167343212	i = 80
maxSIGDs	= 0.0125437855	i = 78	maxSIGDs	= 0.0118727727	i = 75
maxSIGKs	= 0.0287562243	i = 60	maxSIGKs	= 0.0086185896	i = 30
maxSIGNs	= 0.0228258924	i = 45	maxSIGNs	= 0.0285777451	i = 67
maxSIMon	= 0.9440782954	i = 45	maxSIMon	= 0.9347370376	i = 45

A.1.6 Pearson's Correlation with Z-score, MI, Various Noise Levels, and Relaxed Module

20% Standard Deviation, OF 1

10000 runs

maxIDS	= 0.0201641188	i = 59
maxMDS	= 0.0673452936	i = 29
maxMUB	= 0.1132810352	i = 68
maxMUD	= 0.0244435856	i = 47
maxMUK	= 0.0148392212	i = 18
maxMUN	= 0.0252303148	i = 35
maxMUP	= 0.0165728408	i = 20
maxMUS	= 0.0183391302	i = 53
maxMUBs	= 0.0460302978	i = 34
maxMUDs	= 0.0192201500	i = 53
maxMUKs	= 0.0454539297	i = 30
maxMUNs	= 0.1619543925	i = 23
maxSIGB	= 0.1170433644	i = 46
maxSIGD	= 0.0228406448	i = 49
maxSIGK	= 0.0794527425	i = 57
maxSIGN	= 0.0380564974	i = 39
maxSIGP	= 0.0200674377	i = 65
maxSIGS	= 0.0158796055	i = 76
maxSIGBs	= 0.0356888728	i = 67
maxSIGDs	= 0.0158280776	i = 37
maxSIGKs	= 0.0609497325	i = 39
maxSIGNs	= 0.0305724770	i = 36
maxSIMon	= 0.5851482953	i = 45

20% Standard Deviation, RM, OF 2/3

10000 runs

maxIDS	= 0.0278658600	i = 59
maxMDS	= 0.0638996413	i = 29
maxMUB	= 0.1004479950	i = 60
maxMUD	= 0.0252531657	i = 47
maxMUK	= 0.0159209534	i = 18
maxMUN	= 0.0178290575	i = 35
maxMUP	= 0.0159334519	i = 20
maxMUS	= 0.0149020717	i = 53
maxMUBs	= 0.0459119646	i = 35
maxMUDs	= 0.0207787032	i = 53
maxMUKs	= 0.0432131018	i = 30
maxMUNs	= 0.1545240196	i = 23
maxSIGB	= 0.1056536705	i = 39
maxSIGD	= 0.0177141323	i = 89
maxSIGK	= 0.0740829709	i = 57
maxSIGN	= 0.0386100979	i = 38
maxSIGP	= 0.0237570040	i = 65
maxSIGS	= 0.0160479707	i = 80
maxSIGBs	= 0.0310335689	i = 70
maxSIGDs	= 0.0213193903	i = 37
maxSIGKs	= 0.0573993043	i = 39
maxSIGNs	= 0.0353307154	i = 36
maxSIMon	= 0.6002422270	i = 45

20% Standard Deviation, RM, OF 4

```
               10000 runs
maxIDS    = 0.0360040022       i = 26
maxMDS    = 0.0240018208       i = 29
maxMUB    = 0.0871837822       i = 68
maxMUD    = 0.0165354425       i = 63
maxMUK    = 0.0199805312       i = 27
maxMUN    = 0.0330358894       i = 54
maxMUP    = 0.0163367379       i = 21
maxMUS    = 0.0086054767       i = 90
maxMUBs   = 0.0315483588       i = 35
maxMUDs   = 0.0136746047       i = 51
maxMUKs   = 0.0332272220       i = 25
maxMUNs   = 0.0426586997       i = 31
maxSIGB   = 0.1195160285       i = 38
maxSIGD   = 0.0221313708       i = 49
maxSIGK   = 0.0554584958       i = 56
maxSIGN   = 0.0213230794       i = 39
maxSIGP   = 0.0162398006       i = 60
maxSIGS   = 0.0153712624       i = 64
maxSIGBs  = 0.0165851822       i = 70
maxSIGDs  = 0.0200294235       i = 37
maxSIGKs  = 0.0471823010       i = 39
maxSIGNs  = 0.0281700417       i = 39
maxSIMon  = 0.7072672839       i = 45
```

25% Standard Deviation, RM, OF 1

```
               10000 runs
maxIDS    = 0.0588055294       i = 33
maxMDS    = 0.0963706598       i = 49
maxMUB    = 0.2061688676       i = 63
maxMUD    = 0.0236645443       i = 82
maxMUK    = 0.0462917733       i = 23
maxMUN    = 0.0886925616       i = 34
maxMUP    = 0.0173255514       i = 17
maxMUS    = 0.0209470309       i = 53
maxMUBs   = 0.0539446054       i = 42
maxMUDs   = 0.0325947199       i = 47
maxMUKs   = 0.0185678031       i = 15
maxMUNs   = 0.1467569510       i = 25
maxSIGB   = 0.2038303463       i = 44
maxSIGD   = 0.0221864753       i = 72
maxSIGK   = 0.1056587724       i = 61
maxSIGN   = 0.0342440372       i = 34
maxSIGP   = 0.0348813053       i = 38
maxSIGS   = 0.0196275771       i = 17
maxSIGBs  = 0.0317398823       i = 21
maxSIGDs  = 0.0177312132       i = 90
maxSIGKs  = 0.0597123029       i = 36
maxSIGNs  = 0.0404665252       i = 42
maxSIMon  = 0.5148536810       i = 45
```

25% Standard Deviation, RM, OF 2/3

10000 runs

maxIDS	= 0.0659622681	i = 46
maxMDS	= 0.0848123474	i = 49
maxMUB	= 0.2023292528	i = 66
maxMUD	= 0.0186176000	i = 85
maxMUK	= 0.0401888353	i = 19
maxMUN	= 0.0943058129	i = 34
maxMUP	= 0.0142038735	i = 36
maxMUS	= 0.0194015417	i = 37
maxMUBs	= 0.0554827932	i = 39
maxMUDs	= 0.0273443572	i = 75
maxMUKs	= 0.0196887617	i = 22
maxMUNs	= 0.1349927215	i = 25
maxSIGB	= 0.1975779583	i = 44
maxSIGD	= 0.0234357999	i = 27
maxSIGK	= 0.1007557603	i = 61
maxSIGN	= 0.0335807480	i = 34
maxSIGP	= 0.0355308028	i = 38
maxSIGS	= 0.0208785282	i = 83
maxSIGBs	= 0.0321699910	i = 57
maxSIGDs	= 0.0165972608	i = 85
maxSIGKs	= 0.0569236238	i = 36
maxSIGNs	= 0.0390352247	i = 42
maxSIMon	= 0.5190521470	i = 45

25% Standard Deviation, RM, OF 4

10000 runs

maxIDS	= 0.0824316746	i = 46
maxMDS	= 0.0305050006	i = 43
maxMUB	= 0.1842904443	i = 66
maxMUD	= 0.0179583723	i = 48
maxMUK	= 0.0274775235	i = 19
maxMUN	= 0.1082209955	i = 46
maxMUP	= 0.0195892414	i = 17
maxMUS	= 0.0250005375	i = 71
maxMUBs	= 0.0498993976	i = 34
maxMUDs	= 0.0188076246	i = 27
maxMUKs	= 0.0133896097	i = 14
maxMUNs	= 0.0546222440	i = 19
maxSIGB	= 0.1840857458	i = 44
maxSIGD	= 0.0197657210	i = 26
maxSIGK	= 0.0975847371	i = 38
maxSIGN	= 0.0101166737	i = 12
maxSIGP	= 0.0334191544	i = 49
maxSIGS	= 0.0152318961	i = 83
maxSIGBs	= 0.0247376097	i = 55
maxSIGDs	= 0.0111450931	i = 26
maxSIGKs	= 0.0543593162	i = 44
maxSIGNs	= 0.0415340830	i = 42
maxSIMon	= 0.5854936414	i = 45

33% Standard Deviation, RM, OF 1

10000 runs

```
maxIDS   = 0.1337638510      i = 26
maxMDS   = 0.0648856061      i = 39
maxMUB   = 0.2943183032      i = 57
maxMUD   = 0.0335818089      i = 30
maxMUK   = 0.0627534116      i = 57
maxMUN   = 0.1244680029      i = 56
maxMUP   = 0.0434910071      i = 48
maxMUS   = 0.0457534684      i = 52
maxMUBs  = 0.0889031307      i = 48
maxMUDs  = 0.0279854356      i = 79
maxMUKs  = 0.0654594428      i = 50
maxMUNs  = 0.1353255229      i = 26
maxSIGB  = 0.3527848923      i = 53
maxSIGD  = 0.0574721258      i = 36
maxSIGK  = 0.0946861093      i = 77
maxSIGN  = 0.0286143361      i = 82
maxSIGP  = 0.0347901545      i = 51
maxSIGS  = 0.0392271279      i = 57
maxSIGBs = 0.0501070813      i = 40
maxSIGDs = 0.0351452653      i = 59
maxSIGKs = 0.0349584051      i = 24
maxSIGNs = 0.0648348760      i = 30
maxSIMon = 0.4633459598      i = 45
```

33% Standard Deviation, RM, OF 2/3

10000 runs

```
maxIDS   = 0.1410931994      i = 26
maxMDS   = 0.0517260656      i = 46
maxMUB   = 0.2809330042      i = 57
maxMUD   = 0.0269332729      i = 30
maxMUK   = 0.0565930224      i = 59
maxMUN   = 0.1435426149      i = 56
maxMUP   = 0.0498567619      i = 48
maxMUS   = 0.0392524354      i = 52
maxMUBs  = 0.0885094706      i = 48
maxMUDs  = 0.0239179664      i = 6
maxMUKs  = 0.0614452679      i = 50
maxMUNs  = 0.1243135997      i = 26
maxSIGB  = 0.3543105300      i = 53
maxSIGD  = 0.0530481125      i = 36
maxSIGK  = 0.0892670976      i = 77
maxSIGN  = 0.0282729733      i = 76
maxSIGP  = 0.0436942384      i = 51
maxSIGS  = 0.0376062517      i = 57
maxSIGBs = 0.0506516573      i = 40
maxSIGDs = 0.0440379902      i = 59
maxSIGKs = 0.0443758575      i = 51
maxSIGNs = 0.0691921882      i = 29
maxSIMon = 0.4650858987      i = 45
```

33% Standard Deviation, RM, OF 4

```
             10000 runs
maxIDS   = 0.1569628569      i = 26
maxMDS   = 0.0329242816      i = 79
maxMUB   = 0.2624640328      i = 62
maxMUD   = 0.0212190091      i = 29
maxMUK   = 0.0709043371      i = 63
maxMUN   = 0.1543942400      i = 56
maxMUP   = 0.0310296803      i = 28
maxMUS   = 0.0176492228      i = 84
maxMUBs  = 0.0608584830      i = 50
maxMUDs  = 0.0292638383      i = 79
maxMUKs  = 0.0525732154      i = 24
maxMUNs  = 0.0653637451      i = 44
maxSIGB  = 0.3248045618      i = 53
maxSIGD  = 0.0254089422      i = 22
maxSIGK  = 0.0847118025      i = 47
maxSIGN  = 0.0241778455      i = 82
maxSIGP  = 0.0286476331      i = 53
maxSIGS  = 0.0256638331      i = 60
maxSIGBs = 0.0290509913      i = 80
maxSIGDs = 0.0306972711      i = 68
maxSIGKs = 0.0520792001      i = 34
maxSIGNs = 0.0598743940      i = 54
maxSIMon = 0.4864408561      i = 44
```

A.2 Two SIMs

A.2.1 Pearson's Correlation with Z-score, Noise, and Two Modules

K-S Statistics and Indices for PC, Z-score, Noise, Two Modules, OF 1

5000 runs			10000 runs		
maxDCO	= 0.0419949800	i = 26	maxDCO	= 0.0167064680	i = 31
maxDSG	= 0.0177891858	i = 30	maxDSG	= 0.0194349099	i = 66
maxDSO	= 0.0332819600	i = 37	maxDSO	= 0.0122076266	i = 63
maxGT	= 0.0276649747	i = 90	maxGT	= 0.0102707661	i = 9
maxPC	= 0.0392061113	i = 23	maxPC	= 0.0115764079	i = 85
maxSOGCO	= 0.0108119531	i = 50	maxSOGCO	= 0.0078421195	i = 78
maxSUPP	= 0.1713234226	i = 59	maxSUPP	= 0.1548458385	i = 61
maxZPC	= 0.2372577246	i = 33	maxZPC	= 0.2541706867	i = 26
maxIDS	= 0.0369206477	i = 53	maxIDS	= 0.0289143530	i = 26
maxMDS	= 0.0806423260	i = 53	maxMDS	= 0.0408183600	i = 29
maxMUB	= 0.0373346091	i = 34	maxMUB	= 0.0762313609	i = 69
maxMUD	= 0.0466330416	i = 29	maxMUD	= 0.0117162538	i = 63
maxMUK	= 0.0262625875	i = 79	maxMUK	= 0.0175370678	i = 64
maxMUN	= 0.0190562713	i = 19	maxMUN	= 0.0344826071	i = 36
maxMUP	= 0.0341757778	i = 62	maxMUP	= 0.0163270856	i = 60
maxMUS	= 0.0266079684	i = 72	maxMUS	= 0.0151413929	i = 58
maxMUBs	= 0.0320442071	i = 57	maxMUBs	= 0.0173806327	i = 49
maxMUDs	= 0.0254956021	i = 75	maxMUDs	= 0.0150595836	i = 84
maxMUKs	= 0.0201192680	i = 18	maxMUKs	= 0.0150503329	i = 14
maxMUNs	= 0.2161188391	i = 34	maxMUNs	= 0.2075656215	i = 27
maxSIGB	= 0.0284612926	i = 75	maxSIGB	= 0.0155694061	i = 56
maxSIGD	= 0.0178061244	i = 51	maxSIGD	= 0.0285954361	i = 73
maxSIGK	= 0.0366163737	i = 46	maxSIGK	= 0.0197592015	i = 42
maxSIGN	= 0.0201686376	i = 16	maxSIGN	= 0.0117079850	i = 64
maxSIGP	= 0.0163427997	i = 33	maxSIGP	= 0.0161529242	i = 86
maxSIGS	= 0.0226143976	i = 56	maxSIGS	= 0.0240910133	i = 14
maxSIGBs	= 0.0383364618	i = 65	maxSIGBs	= 0.0146644905	i = 35
maxSIGDs	= 0.0202996015	i = 62	maxSIGDs	= 0.0173893666	i = 47
maxSIGKs	= 0.0291508152	i = 53	maxSIGKs	= 0.0156148328	i = 63
maxSIGNs	= 0.0382959746	i = 43	maxSIGNs	= 0.0226344145	i = 45
maxSIMon	= 0.5982116206	i = 45	maxSIMon	= 0.6087420945	i = 45

K-S Statistics and Indices for PC, Z-score, Noise, Two Modules, OF 2

5000 runs			10000 runs		
maxDCO	= 0.0433488353	i = 26	maxDCO	= 0.0192683123	i = 31
maxDSG	= 0.0199565672	i = 32	maxDSG	= 0.0198148035	i = 71
maxDSO	= 0.0323267296	i = 37	maxDSO	= 0.0114721016	i = 63
maxGT	= 0.0231894450	i = 64	maxGT	= 0.0093467780	i = 9
maxPC	= 0.0379381697	i = 23	maxPC	= 0.0104941321	i = 40
maxSOGCO	= 0.0124582574	i = 50	maxSOGCO	= 0.0086921603	i = 71
maxSUPP	= 0.1759070701	i = 59	maxSUPP	= 0.1567985790	i = 61
maxZPC	= 0.2328409580	i = 33	maxZPC	= 0.2529636639	i = 26
maxIDS	= 0.0368789822	i = 23	maxIDS	= 0.0290059698	i = 36
maxMDS	= 0.0804572740	i = 53	maxMDS	= 0.0392452028	i = 29
maxMUB	= 0.0328983221	i = 34	maxMUB	= 0.0732184150	i = 69
maxMUD	= 0.0441130073	i = 29	maxMUD	= 0.0109882045	i = 63
maxMUK	= 0.0244985761	i = 79	maxMUK	= 0.0149383175	i = 64
maxMUN	= 0.0185040257	i = 19	maxMUN	= 0.0341065203	i = 36
maxMUP	= 0.0304972240	i = 62	maxMUP	= 0.0166739950	i = 60
maxMUS	= 0.0291122903	i = 72	maxMUS	= 0.0167106881	i = 58
maxMUBs	= 0.0295158878	i = 57	maxMUBs	= 0.0170631272	i = 49
maxMUDs	= 0.0254553277	i = 75	maxMUDs	= 0.0152426546	i = 84
maxMUKs	= 0.0188912335	i = 16	maxMUKs	= 0.0162417972	i = 14
maxMUNs	= 0.2101124746	i = 34	maxMUNs	= 0.2051185990	i = 28
maxSIGB	= 0.0297146135	i = 33	maxSIGB	= 0.0155885156	i = 56
maxSIGD	= 0.0181577923	i = 53	maxSIGD	= 0.0285238200	i = 73
maxSIGK	= 0.0365286513	i = 38	maxSIGK	= 0.0179393427	i = 42
maxSIGN	= 0.0180635512	i = 16	maxSIGN	= 0.0123755543	i = 18
maxSIGP	= 0.0186535822	i = 32	maxSIGP	= 0.0170470932	i = 86
maxSIGS	= 0.0251111430	i = 56	maxSIGS	= 0.0237794087	i = 14
maxSIGBs	= 0.0387371699	i = 65	maxSIGBs	= 0.0144054963	i = 35
maxSIGDs	= 0.0188707463	i = 62	maxSIGDs	= 0.0171935574	i = 47
maxSIGKs	= 0.0326155989	i = 53	maxSIGKs	= 0.0177447766	i = 63
maxSIGNs	= 0.0334268915	i = 42	maxSIGNs	= 0.0232978756	i = 45
maxSIMon	= 0.6013009363	i = 45	maxSIMon	= 0.6103168128	i = 45

K-S Statistics and Indices for PC, Z-score, Noise, Two Modules, OF 3

5000 runs			10000 runs		
maxDCO	= 0.0316732098	i = 19	maxDCO	= 0.0163516477	i = 86
maxDSG	= 0.0175458585	i = 32	maxDSG	= 0.0193142058	i = 87
maxDSO	= 0.0180530498	i = 91	maxDSO	= 0.0142675673	i = 61
maxGT	= 0.0313042182	i = 64	maxGT	= 0.0090194371	i = 69
maxPC	= 0.0250095045	i = 85	maxPC	= 0.0125507365	i = 52
maxSOGCO	= 0.0124133649	i = 82	maxSOGCO	= 0.0150191508	i = 35
maxSUPP	= 0.0799799533	i = 59	maxSUPP	= 0.0618965927	i = 66
maxZPC	= 0.1946361638	i = 23	maxZPC	= 0.2233462478	i = 26
maxIDS	= 0.0361238263	i = 76	maxIDS	= 0.0211508014	i = 26
maxMDS	= 0.0390535573	i = 49	maxMDS	= 0.0067063909	i = 29
maxMUB	= 0.0362516997	i = 35	maxMUB	= 0.0610788452	i = 69
maxMUD	= 0.0235521255	i = 29	maxMUD	= 0.0176725811	i = 47
maxMUK	= 0.0222389699	i = 9	maxMUK	= 0.0148785631	i = 29
maxMUN	= 0.0246038547	i = 46	maxMUN	= 0.0456430173	i = 36
maxMUP	= 0.0399030640	i = 62	maxMUP	= 0.0202893212	i = 52
maxMUS	= 0.0374357771	i = 71	maxMUS	= 0.0220269750	i = 58
maxMUBs	= 0.0269186572	i = 16	maxMUBs	= 0.0145718697	i = 71
maxMUDs	= 0.0219243633	i = 48	maxMUDs	= 0.0203162302	i = 63
maxMUKs	= 0.0141989467	i = 18	maxMUKs	= 0.0122501726	i = 12
maxMUNs	= 0.1670550773	i = 34	maxMUNs	= 0.1787455532	i = 29
maxSIGB	= 0.0269029957	i = 75	maxSIGB	= 0.0265804262	i = 55
maxSIGD	= 0.0157508109	i = 16	maxSIGD	= 0.0181817660	i = 69
maxSIGK	= 0.0399918698	i = 38	maxSIGK	= 0.0155277706	i = 46
maxSIGN	= 0.0246186557	i = 23	maxSIGN	= 0.0079820267	i = 18
maxSIGP	= 0.0129639264	i = 38	maxSIGP	= 0.0136870452	i = 39
maxSIGS	= 0.0155124465	i = 71	maxSIGS	= 0.0184538559	i = 14
maxSIGBs	= 0.0514268219	i = 65	maxSIGBs	= 0.0211431069	i = 35
maxSIGDs	= 0.0186562752	i = 20	maxSIGDs	= 0.0149851124	i = 7
maxSIGKs	= 0.0300975952	i = 53	maxSIGKs	= 0.0151414369	i = 64
maxSIGNs	= 0.0275740302	i = 44	maxSIGNs	= 0.0281871430	i = 42
maxSIMon	= 0.6969515687	i = 45	maxSIMon	= 0.6972346430	i = 45

K-S Statistics and Indices for PC, Z-score, Noise, Two Modules, OF 4

5000 runs			10000 runs		
maxDCO	= 0.0317973756	i = 61	maxDCO	= 0.0200946562	i = 87
maxDSG	= 0.0207750992	i = 49	maxDSG	= 0.0270733355	i = 85
maxDSO	= 0.0177601465	i = 64	maxDSO	= 0.0200363406	i = 63
maxGT	= 0.0245793917	i = 64	maxGT	= 0.0125841387	i = 69
maxPC	= 0.0220364154	i = 83	maxPC	= 0.0111264532	i = 23
maxSOGCO	= 0.0212104567	i = 24	maxSOGCO	= 0.0190982165	i = 35
maxSUPP	= 0.1009785373	i = 59	maxSUPP	= 0.0776832165	i = 53
maxZPC	= 0.2401708880	i = 33	maxZPC	= 0.2664176324	i = 29
maxIDS	= 0.0416844675	i = 76	maxIDS	= 0.0189783933	i = 29
maxMDS	= 0.0350645916	i = 49	maxMDS	= 0.0119858820	i = 49
maxMUB	= 0.0299623640	i = 26	maxMUB	= 0.0264544535	i = 81
maxMUD	= 0.0258813956	i = 58	maxMUD	= 0.0135991985	i = 47
maxMUK	= 0.0195381955	i = 66	maxMUK	= 0.0234312499	i = 29
maxMUN	= 0.0169830129	i = 40	maxMUN	= 0.0390193174	i = 34
maxMUP	= 0.0339904384	i = 62	maxMUP	= 0.0141915213	i = 87
maxMUS	= 0.0277408199	i = 71	maxMUS	= 0.0143228030	i = 58
maxMUBs	= 0.0255233445	i = 57	maxMUBs	= 0.0139315769	i = 35
maxMUDs	= 0.0203967043	i = 81	maxMUDs	= 0.0204212645	i = 64
maxMUKs	= 0.0278750890	i = 34	maxMUKs	= 0.0101363566	i = 12
maxMUNs	= 0.0259424270	i = 60	maxMUNs	= 0.0158513266	i = 23
maxSIGB	= 0.0315329061	i = 68	maxSIGB	= 0.0399863742	i = 55
maxSIGD	= 0.0235784762	i = 18	maxSIGD	= 0.0143105670	i = 74
maxSIGK	= 0.0410131218	i = 35	maxSIGK	= 0.0099273208	i = 46
maxSIGN	= 0.0191475943	i = 22	maxSIGN	= 0.0088904080	i = 22
maxSIGP	= 0.0110385515	i = 3	maxSIGP	= 0.0126958182	i = 14
maxSIGS	= 0.0123446241	i = 71	maxSIGS	= 0.0184468006	i = 12
maxSIGBs	= 0.0545702370	i = 65	maxSIGBs	= 0.0239248659	i = 35
maxSIGDs	= 0.0219265588	i = 13	maxSIGDs	= 0.0083192015	i = 26
maxSIGKs	= 0.0206693114	i = 53	maxSIGKs	= 0.0157867497	i = 45
maxSIGNs	= 0.0322408707	i = 42	maxSIGNs	= 0.0273235402	i = 41
maxSIMon	= 0.7829885057	i = 45	maxSIMon	= 0.7781594278	i = 45

A.2.2 Pearson's Correlation with Z-score, MI, Noise, and Two Modules

K-S Statistics and Indices for PC, Z-score, MI, Noise, Two Modules, OF 1

	5000 runs			10000 runs	
maxDCO	= 0.0173685056	i = 16	maxDCO	= 0.0153912146	i = 76
maxDSG	= 0.0146046730	i = 10	maxDSG	= 0.0141790141	i = 19
maxDSO	= 0.0161136286	i = 19	maxDSO	= 0.0114928824	i = 52
maxGT	= 0.0161458248	i = 38	maxGT	= 0.0108497036	i = 36
maxPC	= 0.0230912875	i = 31	maxPC	= 0.0160595970	i = 23
maxSOGCO	= 0.0219729536	i = 12	maxSOGCO	= 0.0156301649	i = 68
maxSUPP	= 0.1538486902	i = 71	maxSUPP	= 0.1591331870	i = 68
maxZPC	= 0.1731893115	i = 28	maxZPC	= 0.1476994363	i = 26
maxZMI	= 0.0207778142	i = 78	maxZMI	= 0.0318695806	i = 46
maxDIM	= 0.0172975187	i = 42	maxDIM	= 0.0181674979	i = 28
maxMI	= 0.0489506746	i = 33	maxMI	= 0.0177249793	i = 66
maxNEG	= 0.0309044967	i = 24	maxNEG	= 0.0191998798	i = 50
maxPEAK	= 0.0318653406	i = 1	maxPEAK	= 0.0093767418	i = 49
maxSFACT	= 0.0297493117	i = 45	maxSFACT	= 0.0134994838	i = 59
maxIDS	= 0.0360803307	i = 39	maxIDS	= 0.0449691982	i = 43
maxMDS	= 0.0471224721	i = 29	maxMDS	= 0.0539955118	i = 36
maxMUB	= 0.0817512574	i = 53	maxMUB	= 0.0730412613	i = 47
maxMUD	= 0.0275828549	i = 37	maxMUD	= 0.0143951842	i = 66
maxMUK	= 0.0272360666	i = 13	maxMUK	= 0.0251377721	i = 46
maxMUN	= 0.0320340799	i = 47	maxMUN	= 0.0354281477	i = 40
maxMUP	= 0.0323831956	i = 68	maxMUP	= 0.0146234714	i = 62
maxMUS	= 0.0201463027	i = 47	maxMUS	= 0.0136952971	i = 24
maxMUBs	= 0.0210761142	i = 52	maxMUBs	= 0.0099636970	i = 96
maxMUDs	= 0.0170787396	i = 50	maxMUDs	= 0.0103984606	i = 67
maxMUKs	= 0.0352284885	i = 43	maxMUKs	= 0.0143840364	i = 19
maxMUNs	= 0.1918243312	i = 38	maxMUNs	= 0.1863517175	i = 24
maxSIGB	= 0.0524810100	i = 24	maxSIGB	= 0.0149346400	i = 62
maxSIGD	= 0.0229787947	i = 85	maxSIGD	= 0.0222747299	i = 40
maxSIGK	= 0.0239815326	i = 62	maxSIGK	= 0.0328660957	i = 40
maxSIGN	= 0.0273532143	i = 29	maxSIGN	= 0.0170784077	i = 62
maxSIGP	= 0.0258586112	i = 48	maxSIGP	= 0.0278786733	i = 48
maxSIGS	= 0.0312113306	i = 42	maxSIGS	= 0.0185188955	i = 32
maxSIGBs	= 0.0394283502	i = 29	maxSIGBs	= 0.0205676646	i = 41
maxSIGDs	= 0.0171625273	i = 44	maxSIGDs	= 0.0151367542	i = 74
maxSIGKs	= 0.0292500762	i = 78	maxSIGKs	= 0.0190966416	i = 55
maxSIGNs	= 0.0175391844	i = 47	maxSIGNs	= 0.0140500875	i = 62
maxSIMon	= 0.6372284559	i = 44	maxSIMon	= 0.6207717176	i = 44

K-S Statistics and Indices for PC, Z-score, MI, Noise, Two Modules, OF 2

5000 runs			10000 runs		
maxDCO	= 0.0188804556	i = 16	maxDCO	= 0.0171235129	i = 63
maxDSG	= 0.0113790814	i = 10	maxDSG	= 0.0150507982	i = 19
maxDSO	= 0.0148896287	i = 19	maxDSO	= 0.0116506609	i = 77
maxGT	= 0.0156578844	i = 38	maxGT	= 0.0096629760	i = 36
maxPC	= 0.0208031081	i = 39	maxPC	= 0.0166200408	i = 22
maxSOGCO	= 0.0214170225	i = 12	maxSOGCO	= 0.0171032974	i = 68
maxSUPP	= 0.1590634527	i = 70	maxSUPP	= 0.1634075510	i = 60
maxZPC	= 0.1715179388	i = 28	maxZPC	= 0.1472470542	i = 26
maxZMI	= 0.0232290478	i = 78	maxZMI	= 0.0337162005	i = 46
maxDIM	= 0.0242666820	i = 42	maxDIM	= 0.0222225693	i = 28
maxMI	= 0.0456252273	i = 33	maxMI	= 0.0229283404	i = 66
maxNEG	= 0.0305672611	i = 24	maxNEG	= 0.0157876112	i = 50
maxPEAK	= 0.0329126748	i = 1	maxPEAK	= 0.0118652712	i = 11
maxSFACT	= 0.0277944848	i = 45	maxSFACT	= 0.0134547294	i = 59
maxIDS	= 0.0378619521	i = 39	maxIDS	= 0.0463721289	i = 43
maxMDS	= 0.0433037797	i = 26	maxMDS	= 0.0512658683	i = 36
maxMUB	= 0.0760682484	i = 53	maxMUB	= 0.0616325555	i = 47
maxMUD	= 0.0294308868	i = 37	maxMUD	= 0.0130442870	i = 66
maxMUK	= 0.0213349628	i = 13	maxMUK	= 0.0228107837	i = 46
maxMUN	= 0.0275663280	i = 47	maxMUN	= 0.0253702520	i = 40
maxMUP	= 0.0321979113	i = 70	maxMUP	= 0.0128335090	i = 8
maxMUS	= 0.0236485495	i = 47	maxMUS	= 0.0094325093	i = 24
maxMUBs	= 0.0231034657	i = 45	maxMUBs	= 0.0095550959	i = 96
maxMUDs	= 0.0177180062	i = 50	maxMUDs	= 0.0096803172	i = 88
maxMUKs	= 0.0349434622	i = 29	maxMUKs	= 0.0142535303	i = 19
maxMUNs	= 0.1909131852	i = 38	maxMUNs	= 0.1863690410	i = 24
maxSIGB	= 0.0560032947	i = 24	maxSIGB	= 0.0168798248	i = 62
maxSIGD	= 0.0226243364	i = 85	maxSIGD	= 0.0231406993	i = 40
maxSIGK	= 0.0258770089	i = 62	maxSIGK	= 0.0296972379	i = 40
maxSIGN	= 0.0283633430	i = 29	maxSIGN	= 0.0177112963	i = 62
maxSIGP	= 0.0255186685	i = 48	maxSIGP	= 0.0267098902	i = 64
maxSIGS	= 0.0322236029	i = 42	maxSIGS	= 0.0180441340	i = 32
maxSIGBs	= 0.0439012054	i = 29	maxSIGBs	= 0.0213491817	i = 41
maxSIGDs	= 0.0180945608	i = 44	maxSIGDs	= 0.0156440426	i = 74
maxSIGKs	= 0.0278475935	i = 78	maxSIGKs	= 0.0193340956	i = 55
maxSIGNs	= 0.0199848113	i = 47	maxSIGNs	= 0.0157935992	i = 62
maxSIMon	= 0.6417153865	i = 44	maxSIMon	= 0.6261751176	i = 44

K-S Statistics and Indices for PC, Z-score, MI, Noise, Two Modules, OF 3

5000 runs			10000 runs		
maxDCO	= 0.0204811433	i = 52	maxDCO	= 0.0222018393	i = 22
maxDSG	= 0.0166964390	i = 23	maxDSG	= 0.0148693177	i = 69
maxDSO	= 0.0133224261	i = 19	maxDSO	= 0.0138806556	i = 77
maxGT	= 0.0221222391	i = 38	maxGT	= 0.0172224803	i = 35
maxPC	= 0.0221355949	i = 39	maxPC	= 0.0143726502	i = 77
maxSOGCO	= 0.0260788995	i = 13	maxSOGCO	= 0.0117638106	i = 68
maxSUPP	= 0.0424564684	i = 72	maxSUPP	= 0.0655507669	i = 60
maxZPC	= 0.1411693017	i = 22	maxZPC	= 0.1214385467	i = 23
maxZMI	= 0.0174760847	i = 78	maxZMI	= 0.0309148829	i = 36
maxDIM	= 0.0187048198	i = 42	maxDIM	= 0.0209695164	i = 28
maxMI	= 0.0457503464	i = 33	maxMI	= 0.0196497426	i = 68
maxNEG	= 0.0246915641	i = 24	maxNEG	= 0.0152738614	i = 84
maxPEAK	= 0.0354580210	i = 1	maxPEAK	= 0.0086109064	i = 0
maxSFACT	= 0.0329972120	i = 54	maxSFACT	= 0.0063632188	i = 49
maxIDS	= 0.0209836558	i = 29	maxIDS	= 0.0346972998	i = 76
maxMDS	= 0.0228334363	i = 49	maxMDS	= 0.0150789328	i = 33
maxMUB	= 0.0655269704	i = 53	maxMUB	= 0.0379990708	i = 47
maxMUD	= 0.0346566721	i = 37	maxMUD	= 0.0114260232	i = 69
maxMUK	= 0.0251773819	i = 13	maxMUK	= 0.0256992064	i = 28
maxMUN	= 0.0447987446	i = 47	maxMUN	= 0.0245563371	i = 38
maxMUP	= 0.0235963873	i = 68	maxMUP	= 0.0143693124	i = 8
maxMUS	= 0.0159134543	i = 47	maxMUS	= 0.0147424806	i = 80
maxMUBs	= 0.0200687825	i = 52	maxMUBs	= 0.0148966878	i = 16
maxMUDs	= 0.0176563883	i = 24	maxMUDs	= 0.0111663409	i = 58
maxMUKs	= 0.0270138066	i = 32	maxMUKs	= 0.0131530110	i = 19
maxMUNs	= 0.1550259604	i = 26	maxMUNs	= 0.1646586345	i = 24
maxSIGB	= 0.0484832802	i = 24	maxSIGB	= 0.0238033250	i = 62
maxSIGD	= 0.0150353094	i = 85	maxSIGD	= 0.0242813007	i = 40
maxSIGK	= 0.0184794404	i = 72	maxSIGK	= 0.0286585170	i = 26
maxSIGN	= 0.0201255447	i = 29	maxSIGN	= 0.0090608444	i = 86
maxSIGP	= 0.0255329805	i = 48	maxSIGP	= 0.0176777589	i = 38
maxSIGS	= 0.0308135361	i = 42	maxSIGS	= 0.0156670565	i = 31
maxSIGBs	= 0.0332793536	i = 23	maxSIGBs	= 0.0132444672	i = 41
maxSIGDs	= 0.0157031002	i = 61	maxSIGDs	= 0.0138359288	i = 70
maxSIGKs	= 0.0221840097	i = 76	maxSIGKs	= 0.0167812206	i = 58
maxSIGNs	= 0.0188667590	i = 89	maxSIGNs	= 0.0200983455	i = 62
maxSIMon	= 0.7479356917	i = 44	maxSIMon	= 0.7342568412	i = 44

K-S Statistics and Indices for PC, Z-score, MI, Noise, Two Modules, OF 4

5000 runs			10000 runs		
maxDCO	= 0.0161824105	i = 67	maxDCO	= 0.0135062035	i = 27
maxDSG	= 0.0160060245	i = 24	maxDSG	= 0.0256571514	i = 69
maxDSO	= 0.0112202449	i = 55	maxDSO	= 0.0124759843	i = 13
maxGT	= 0.0208844269	i = 33	maxGT	= 0.0238995768	i = 33
maxPC	= 0.0297351391	i = 39	maxPC	= 0.0241065605	i = 76
maxSOGCO	= 0.0273309735	i = 20	maxSOGCO	= 0.0117790760	i = 69
maxSUPP	= 0.0627708713	i = 61	maxSUPP	= 0.0846765287	i = 65
maxZPC	= 0.1793813603	i = 22	maxZPC	= 0.1642915750	i = 23
maxZMI	= 0.0238346643	i = 76	maxZMI	= 0.0384858074	i = 36
maxDIM	= 0.0158376926	i = 42	maxDIM	= 0.0154926935	i = 28
maxMI	= 0.0357330692	i = 25	maxMI	= 0.0202215478	i = 69
maxNEG	= 0.0238797676	i = 28	maxNEG	= 0.0117411979	i = 19
maxPEAK	= 0.0415586403	i = 1	maxPEAK	= 0.0115231680	i = 0
maxSFACT	= 0.0278754344	i = 55	maxSFACT	= 0.0093121411	i = 9
maxIDS	= 0.0222036977	i = 86	maxIDS	= 0.0208469175	i = 76
maxMDS	= 0.0170401782	i = 49	maxMDS	= 0.0073068125	i = 13
maxMUB	= 0.0304044395	i = 53	maxMUB	= 0.0170983542	i = 36
maxMUD	= 0.0374421611	i = 43	maxMUD	= 0.0159023631	i = 66
maxMUK	= 0.0161252260	i = 13	maxMUK	= 0.0196454637	i = 66
maxMUN	= 0.0410713639	i = 47	maxMUN	= 0.0156525205	i = 38
maxMUP	= 0.0297963507	i = 53	maxMUP	= 0.0138994848	i = 62
maxMUS	= 0.0146883646	i = 69	maxMUS	= 0.0145058151	i = 80
maxMUBs	= 0.0142405535	i = 27	maxMUBs	= 0.0134911005	i = 16
maxMUDs	= 0.0156749987	i = 35	maxMUDs	= 0.0120953960	i = 45
maxMUKs	= 0.0152537663	i = 34	maxMUKs	= 0.0131845010	i = 76
maxMUNs	= 0.0148091769	i = 71	maxMUNs	= 0.0133522812	i = 41
maxSIGB	= 0.0482290924	i = 24	maxSIGB	= 0.0420238121	i = 62
maxSIGD	= 0.0157015774	i = 15	maxSIGD	= 0.0177532471	i = 40
maxSIGK	= 0.0301765068	i = 70	maxSIGK	= 0.0212352180	i = 65
maxSIGN	= 0.0219516026	i = 29	maxSIGN	= 0.0090572775	i = 33
maxSIGP	= 0.0187017506	i = 71	maxSIGP	= 0.0179172516	i = 39
maxSIGS	= 0.0335350899	i = 42	maxSIGS	= 0.0154539719	i = 31
maxSIGBs	= 0.0335004571	i = 23	maxSIGBs	= 0.0115058156	i = 41
maxSIGDs	= 0.0253971706	i = 49	maxSIGDs	= 0.0190131449	i = 44
maxSIGKs	= 0.0214441907	i = 81	maxSIGKs	= 0.0184579881	i = 57
maxSIGNs	= 0.0150322770	i = 9	maxSIGNs	= 0.0191407173	i = 63
maxSIMon	= 0.8319508374	i = 44	maxSIMon	= 0.8289067605	i = 44

A.2.3 Pearson's Correlation with Z-score, Noise, and Two Relaxed Modules

K-S Statistics and Indices for PC, Z-score, Noise, RMs, OF 1

5000 runs		10000 runs	
maxIDS = 0.0511475898	i = 29	maxIDS = 0.0476818749	i = 43
maxMDS = 0.0681350725	i = 46	maxMDS = 0.0647352778	i = 29
maxMUB = 0.1236790567	i = 45	maxMUB = 0.1087961174	i = 55
maxMUD = 0.0307161805	i = 46	maxMUD = 0.0194056985	i = 13
maxMUK = 0.0374348026	i = 48	maxMUK = 0.0113383825	i = 20
maxMUN = 0.0713063489	i = 53	maxMUN = 0.0623415014	i = 23
maxMUP = 0.0188172043	i = 22	maxMUP = 0.0244630816	i = 19
maxMUS = 0.0378701751	i = 82	maxMUS = 0.0213181425	i = 77
maxMUBs = 0.0433940541	i = 33	maxMUBs = 0.0356865380	i = 21
maxMUDs = 0.0315066492	i = 60	maxMUDs = 0.0165722071	i = 54
maxMUKs = 0.0387300719	i = 55	maxMUKs = 0.0616118304	i = 44
maxMUNs = 0.2445657991	i = 26	maxMUNs = 0.2301913887	i = 25
maxSIGB = 0.0646672799	i = 26	maxSIGB = 0.0525409150	i = 30
maxSIGD = 0.0260095290	i = 75	maxSIGD = 0.0203820690	i = 85
maxSIGK = 0.0467389009	i = 30	maxSIGK = 0.0437971107	i = 48
maxSIGN = 0.0246419387	i = 66	maxSIGN = 0.0080379471	i = 40
maxSIGP = 0.0205861763	i = 19	maxSIGP = 0.0130160112	i = 81
maxSIGS = 0.0235404896	i = 27	maxSIGS = 0.0122900283	i = 57
maxSIGBs = 0.0594088191	i = 31	maxSIGBs = 0.0603950897	i = 70
maxSIGDs = 0.0250266865	i = 42	maxSIGDs = 0.0140020347	i = 22
maxSIGKs = 0.0723687446	i = 74	maxSIGKs = 0.1035082448	i = 54
maxSIGNs = 0.0384812877	i = 56	maxSIGNs = 0.0309701083	i = 55
maxSIMon = 0.6479015335	i = 45	maxSIMon = 0.6607234029	i = 45

K-S Statistics and Indices for PC, Z-score, Noise, RMs, OF 2

5000 runs			10000 runs		
maxIDS	= 0.0514249110	i = 29	maxIDS	= 0.0470427581	i = 43
maxMDS	= 0.0676395562	i = 46	maxMDS	= 0.0632751532	i = 29
maxMUB	= 0.1232214707	i = 45	maxMUB	= 0.1071574903	i = 55
maxMUD	= 0.0302357431	i = 46	maxMUD	= 0.0188844194	i = 13
maxMUK	= 0.0369667666	i = 48	maxMUK	= 0.0114483435	i = 20
maxMUN	= 0.0708812562	i = 53	maxMUN	= 0.0624859919	i = 23
maxMUP	= 0.0185968925	i = 22	maxMUP	= 0.0240761984	i = 19
maxMUS	= 0.0377009614	i = 82	maxMUS	= 0.0211752526	i = 77
maxMUBs	= 0.0436887527	i = 33	maxMUBs	= 0.0362471058	i = 21
maxMUDs	= 0.0311364061	i = 60	maxMUDs	= 0.0171461992	i = 54
maxMUKs	= 0.0384431092	i = 54	maxMUKs	= 0.0618957586	i = 44
maxMUNs	= 0.2447455451	i = 26	maxMUNs	= 0.2284177138	i = 25
maxSIGB	= 0.0639984051	i = 26	maxSIGB	= 0.0519705128	i = 30
maxSIGD	= 0.0257897249	i = 75	maxSIGD	= 0.0208876646	i = 85
maxSIGK	= 0.0461062457	i = 30	maxSIGK	= 0.0437309462	i = 48
maxSIGN	= 0.0243309148	i = 66	maxSIGN	= 0.0078202393	i = 40
maxSIGP	= 0.0204090633	i = 19	maxSIGP	= 0.0126348149	i = 70
maxSIGS	= 0.0237772863	i = 27	maxSIGS	= 0.0115893626	i = 57
maxSIGBs	= 0.0590999768	i = 31	maxSIGBs	= 0.0602036198	i = 70
maxSIGDs	= 0.0255314373	i = 42	maxSIGDs	= 0.0141386703	i = 22
maxSIGKs	= 0.0725735916	i = 62	maxSIGKs	= 0.1037280867	i = 54
maxSIGNs	= 0.0388658768	i = 56	maxSIGNs	= 0.0303362201	i = 55
maxSIMon	= 0.6481120690	i = 45	maxSIMon	= 0.6608896279	i = 45

K-S Statistics and Indices for PC, Z-score, Noise, RMs, OF 3

	5000 runs			10000 runs	
maxIDS	= 0.0388580887	i = 29	maxIDS	= 0.0238994149	i = 63
maxMDS	= 0.0280250689	i = 59	maxMDS	= 0.0127204799	i = 53
maxMUB	= 0.0936723693	i = 64	maxMUB	= 0.0724085709	i = 58
maxMUD	= 0.0212838826	i = 46	maxMUD	= 0.0188618245	i = 13
maxMUK	= 0.0302465998	i = 64	maxMUK	= 0.0140756619	i = 13
maxMUN	= 0.0455937693	i = 53	maxMUN	= 0.0346991826	i = 23
maxMUP	= 0.0199124893	i = 58	maxMUP	= 0.0165472790	i = 19
maxMUS	= 0.0297774858	i = 81	maxMUS	= 0.0149724319	i = 26
maxMUBs	= 0.0206690308	i = 29	maxMUBs	= 0.0281078122	i = 21
maxMUDs	= 0.0304224568	i = 26	maxMUDs	= 0.0141284107	i = 69
maxMUKs	= 0.0169439708	i = 56	maxMUKs	= 0.0252996077	i = 25
maxMUNs	= 0.2412424898	i = 20	maxMUNs	= 0.2108423799	i = 23
maxSIGB	= 0.0514686248	i = 26	maxSIGB	= 0.0255407450	i = 27
maxSIGD	= 0.0161859719	i = 75	maxSIGD	= 0.0119041296	i = 85
maxSIGK	= 0.0237120375	i = 30	maxSIGK	= 0.0334069588	i = 51
maxSIGN	= 0.0302757474	i = 52	maxSIGN	= 0.0164976092	i = 46
maxSIGP	= 0.0211907723	i = 26	maxSIGP	= 0.0099138554	i = 53
maxSIGS	= 0.0223950529	i = 57	maxSIGS	= 0.0180052037	i = 47
maxSIGBs	= 0.0377647308	i = 18	maxSIGBs	= 0.0324090668	i = 70
maxSIGDs	= 0.0140322900	i = 17	maxSIGDs	= 0.0131671025	i = 64
maxSIGKs	= 0.0324920966	i = 12	maxSIGKs	= 0.0442434724	i = 54
maxSIGNs	= 0.0317077029	i = 59	maxSIGNs	= 0.0132718709	i = 88
maxSIMon	= 0.7869360896	i = 45	maxSIMon	= 0.8005206087	i = 45

K-S Statistics and Indices for PC, Z-score, Noise, RMs, OF 4

5000 runs			10000 runs		
maxIDS	= 0.0363717501	i = 29	maxIDS	= 0.0259086195	i = 63
maxMDS	= 0.0202251351	i = 43	maxMDS	= 0.0246134089	i = 53
maxMUB	= 0.0696399366	i = 76	maxMUB	= 0.0301859394	i = 67
maxMUD	= 0.0204182657	i = 82	maxMUD	= 0.0143713896	i = 13
maxMUK	= 0.0163092682	i = 59	maxMUK	= 0.0107201215	i = 60
maxMUN	= 0.0413023546	i = 62	maxMUN	= 0.0284586180	i = 23
maxMUP	= 0.0174558607	i = 58	maxMUP	= 0.0143322889	i = 12
maxMUS	= 0.0304039509	i = 73	maxMUS	= 0.0143588501	i = 45
maxMUBs	= 0.0264152590	i = 28	maxMUBs	= 0.0336148223	i = 21
maxMUDs	= 0.0378840716	i = 26	maxMUDs	= 0.0146664476	i = 69
maxMUKs	= 0.0239507588	i = 44	maxMUKs	= 0.0176996362	i = 14
maxMUNs	= 0.0294485641	i = 17	maxMUNs	= 0.0137343206	i = 42
maxSIGB	= 0.0569501024	i = 26	maxSIGB	= 0.0369438702	i = 28
maxSIGD	= 0.0171295919	i = 8	maxSIGD	= 0.0126341327	i = 14
maxSIGK	= 0.0262978407	i = 80	maxSIGK	= 0.0392455232	i = 47
maxSIGN	= 0.0303333716	i = 64	maxSIGN	= 0.0128796338	i = 46
maxSIGP	= 0.0227323037	i = 30	maxSIGP	= 0.0122039049	i = 53
maxSIGS	= 0.0240771598	i = 27	maxSIGS	= 0.0138800669	i = 47
maxSIGBs	= 0.0252112406	i = 18	maxSIGBs	= 0.0304422173	i = 70
maxSIGDs	= 0.0121470154	i = 86	maxSIGDs	= 0.0131774161	i = 64
maxSIGKs	= 0.0246847833	i = 91	maxSIGKs	= 0.0338764685	i = 67
maxSIGNs	= 0.0354147592	i = 59	maxSIGNs	= 0.0193142226	i = 24
maxSIMon	= 0.9179538296	i = 45	maxSIMon	= 0.9254414583	i = 44

A.2.4 Pearson's Correlation with Z-score, MI, Noise, and Two Relaxed Modules

K-S Statistics and Indices for PC, Z-score, MI, Noise, RMs, OF 1

5000 runs		10000 runs	
maxIDS = 0.0455364849	i = 23	maxIDS = 0.0267093665	i = 29
maxMDS = 0.0763037945	i = 39	maxMDS = 0.0585380185	i = 49
maxMUB = 0.1366657506	i = 56	maxMUB = 0.1208474683	i = 39
maxMUD = 0.0234843107	i = 63	maxMUD = 0.0124810198	i = 60
maxMUK = 0.0169278717	i = 76	maxMUK = 0.0335760228	i = 63
maxMUN = 0.0621862883	i = 44	maxMUN = 0.0413846601	i = 32
maxMUP = 0.0227910167	i = 21	maxMUP = 0.0191068876	i = 63
maxMUS = 0.0478335330	i = 33	maxMUS = 0.0149274792	i = 75
maxMUBs = 0.0153814371	i = 25	maxMUBs = 0.0267809937	i = 47
maxMUDs = 0.0198900658	i = 77	maxMUDs = 0.0263728852	i = 26
maxMUKs = 0.0495843392	i = 62	maxMUKs = 0.0327972850	i = 53
maxMUNs = 0.2475854552	i = 25	maxMUNs = 0.2435591728	i = 32
maxSIGB = 0.0366954510	i = 49	maxSIGB = 0.0266502763	i = 16
maxSIGD = 0.0150213132	i = 60	maxSIGD = 0.0154465354	i = 75
maxSIGK = 0.0262496394	i = 38	maxSIGK = 0.0345361518	i = 69
maxSIGN = 0.0146137641	i = 56	maxSIGN = 0.0112634887	i = 39
maxSIGP = 0.0179415412	i = 20	maxSIGP = 0.0145360720	i = 40
maxSIGS = 0.0313299286	i = 37	maxSIGS = 0.0197469736	i = 39
maxSIGBs = 0.0284301719	i = 45	maxSIGBs = 0.0427091498	i = 72
maxSIGDs = 0.0159670863	i = 32	maxSIGDs = 0.0122297985	i = 31
maxSIGKs = 0.0583317170	i = 49	maxSIGKs = 0.0670622704	i = 55
maxSIGNs = 0.0480286925	i = 72	maxSIGNs = 0.0333829274	i = 44
maxSIMon = 0.7007296817	i = 45	maxSIMon = 0.6857531853	i = 45

K-S Statistics and Indices for PC, Z-score, MI, Noise, RMs, OF 2

5000 runs			10000 runs		
maxIDS	= 0.0457446177	i = 23	maxIDS	= 0.0269612855	i = 29
maxMDS	= 0.0766421814	i = 39	maxMDS	= 0.0578785319	i = 49
maxMUB	= 0.1361456945	i = 56	maxMUB	= 0.1199019210	i = 39
maxMUD	= 0.0240261322	i = 63	maxMUD	= 0.0124387773	i = 60
maxMUK	= 0.0166460974	i = 55	maxMUK	= 0.0336320630	i = 63
maxMUN	= 0.0617064373	i = 44	maxMUN	= 0.0403262336	i = 32
maxMUP	= 0.0221336231	i = 21	maxMUP	= 0.0190150190	i = 63
maxMUS	= 0.0483899687	i = 33	maxMUS	= 0.0145791947	i = 75
maxMUBs	= 0.0156053539	i = 25	maxMUBs	= 0.0273732480	i = 47
maxMUDs	= 0.0200749908	i = 77	maxMUDs	= 0.0260651665	i = 26
maxMUKs	= 0.0492567433	i = 62	maxMUKs	= 0.0326424595	i = 53
maxMUNs	= 0.2477515369	i = 25	maxMUNs	= 0.2432369208	i = 32
maxSIGB	= 0.0362701335	i = 49	maxSIGB	= 0.0267134247	i = 26
maxSIGD	= 0.0146921876	i = 60	maxSIGD	= 0.0155186828	i = 75
maxSIGK	= 0.0257214764	i = 38	maxSIGK	= 0.0347105075	i = 69
maxSIGN	= 0.0143722212	i = 90	maxSIGN	= 0.0108332453	i = 39
maxSIGP	= 0.0172594231	i = 20	maxSIGP	= 0.0136516221	i = 40
maxSIGS	= 0.0309951227	i = 37	maxSIGS	= 0.0201720535	i = 39
maxSIGBs	= 0.0280321931	i = 45	maxSIGBs	= 0.0419598022	i = 72
maxSIGDs	= 0.0156841516	i = 32	maxSIGDs	= 0.0132804879	i = 31
maxSIGKs	= 0.0587506711	i = 49	maxSIGKs	= 0.0666732553	i = 55
maxSIGNs	= 0.0474034626	i = 72	maxSIGNs	= 0.0332627688	i = 44
maxSIMon	= 0.7009614000	i = 45	maxSIMon	= 0.6863139105	i = 45

K-S Statistics and Indices for PC, Z-score, MI, Noise, RMs, OF 3

5000 runs			10000 runs		
maxIDS	= 0.0304703093	i = 23	maxIDS	= 0.0189020864	i = 39
maxMDS	= 0.0318533310	i = 53	maxMDS	= 0.0103234577	i = 63
maxMUB	= 0.0994693493	i = 57	maxMUB	= 0.0698095183	i = 35
maxMUD	= 0.0286940881	i = 63	maxMUD	= 0.0110767436	i = 79
maxMUK	= 0.0178599898	i = 78	maxMUK	= 0.0347753171	i = 63
maxMUN	= 0.0293324552	i = 15	maxMUN	= 0.0262518010	i = 32
maxMUP	= 0.0123736612	i = 96	maxMUP	= 0.0164594869	i = 76
maxMUS	= 0.0361529968	i = 40	maxMUS	= 0.0146810585	i = 75
maxMUBs	= 0.0116863248	i = 51	maxMUBs	= 0.0162016502	i = 14
maxMUDs	= 0.0200331033	i = 76	maxMUDs	= 0.0240406099	i = 34
maxMUKs	= 0.0318459454	i = 37	maxMUKs	= 0.0183201007	i = 82
maxMUNs	= 0.2197352414	i = 22	maxMUNs	= 0.2092475859	i = 24
maxSIGB	= 0.0202215950	i = 76	maxSIGB	= 0.0137086114	i = 14
maxSIGD	= 0.0183581923	i = 23	maxSIGD	= 0.0216552179	i = 58
maxSIGK	= 0.0124903848	i = 49	maxSIGK	= 0.0189036173	i = 67
maxSIGN	= 0.0174516982	i = 56	maxSIGN	= 0.0126965219	i = 76
maxSIGP	= 0.0211298552	i = 45	maxSIGP	= 0.0095669488	i = 54
maxSIGS	= 0.0285511940	i = 37	maxSIGS	= 0.0210237598	i = 32
maxSIGBs	= 0.0204802492	i = 69	maxSIGBs	= 0.0220450347	i = 75
maxSIGDs	= 0.0161802623	i = 64	maxSIGDs	= 0.0102131520	i = 61
maxSIGKs	= 0.0223955111	i = 74	maxSIGKs	= 0.0180863556	i = 73
maxSIGNs	= 0.0257019831	i = 72	maxSIGNs	= 0.0140515341	i = 26
maxSIMon	= 0.8275882166	i = 45	maxSIMon	= 0.8190395133	i = 45

K-S Statistics and Indices for PC, Z-score, MI, Noise, RMs, OF 4

5000 runs			10000 runs		
maxIDS	= 0.0142394167	i = 23	maxIDS	= 0.0116298094	i = 69
maxMDS	= 0.0199795465	i = 53	maxMDS	= 0.0136630176	i = 23
maxMUB	= 0.0570414925	i = 57	maxMUB	= 0.0196102852	i = 43
maxMUD	= 0.0253888870	i = 63	maxMUD	= 0.0126470364	i = 54
maxMUK	= 0.0260717445	i = 76	maxMUK	= 0.0183512000	i = 85
maxMUN	= 0.0190806765	i = 61	maxMUN	= 0.0138078994	i = 63
maxMUP	= 0.0129188386	i = 41	maxMUP	= 0.0200779259	i = 76
maxMUS	= 0.0366821508	i = 33	maxMUS	= 0.0216699379	i = 37
maxMUBs	= 0.0207243061	i = 57	maxMUBs	= 0.0127072398	i = 14
maxMUDs	= 0.0169469886	i = 76	maxMUDs	= 0.0254817963	i = 36
maxMUKs	= 0.0225729864	i = 37	maxMUKs	= 0.0128256369	i = 36
maxMUNs	= 0.0301166597	i = 35	maxMUNs	= 0.0164754930	i = 84
maxSIGB	= 0.0388874130	i = 49	maxSIGB	= 0.0180056941	i = 29
maxSIGD	= 0.0265540650	i = 60	maxSIGD	= 0.0226949214	i = 23
maxSIGK	= 0.0218817464	i = 49	maxSIGK	= 0.0098797336	i = 49
maxSIGN	= 0.0155309781	i = 55	maxSIGN	= 0.0133036866	i = 13
maxSIGP	= 0.0182914248	i = 20	maxSIGP	= 0.0153874527	i = 66
maxSIGS	= 0.0265044144	i = 37	maxSIGS	= 0.0227259474	i = 32
maxSIGBs	= 0.0313714667	i = 58	maxSIGBs	= 0.0219111932	i = 71
maxSIGDs	= 0.0199047481	i = 32	maxSIGDs	= 0.0090618447	i = 43
maxSIGKs	= 0.0174557464	i = 29	maxSIGKs	= 0.0227770278	i = 55
maxSIGNs	= 0.0324186439	i = 72	maxSIGNs	= 0.0180262709	i = 50
maxSIMon	= 0.9558386573	i = 45	maxSIMon	= 0.9521325778	i = 44